

CAPITULO 2 MODELOS DE DATOS

Se presentan conceptos generales de Modelos de Datos. Se introduce el Modelo Entidad-Relación, un modelo de datos gráfico que sirve para la estructuración de datos.



2.1 DATOS Y MODELOS DE DATOS

2.1.1 Introducción

Un *modelo de datos* es un mecanismo de abstracción que nos permite interpretar una realidad. La mayoría de los modelos de datos informáticos son *estrictamente tipeados*. Los modelos de datos estrictamente tipeados son aquellos en los que cada *dato* debe pertenecer a alguna *categoría*. Una categoría tiene *propiedades*, de forma tal que los datos pertenecientes a una misma categoría tienen las mismas propiedades. La principal ventaja de tales modelos es que las propiedades de los datos pueden ser abstraídas en base a las categorías a las que los mismos pertenecen.

Para una aplicación en particular, los nombres de las categorías junto con sus propiedades se denominan *esquema*. Un esquema es un concepto genérico que identifica categorías (PERSONAS, AUTOS), sus propiedades (Nombre de la persona, marca del auto), y relaciones entre las categorías (PERTENECE, CONDUCE).

Ejemplo:

Un modelo de datos simple es aquél en el que las categorías se denominan *tipos de entidades* y las propiedades se denominan *atributos*. Si aplicamos este modelo de datos a una aplicación que interprete datos de empleados, el esquema podría consistir en *EMPLEADO*(#Empleado, Nombre, Edad, Domicilio), donde el nombre "EMPLEADO" es el nombre genérico de un tipo de entidad y "#Empleado", "Nombre", "Edad" y "Domicilio" son los nombres genéricos de los atributos. Todos los datos en esta aplicación serán de la forma (11024, Juan Pérez, 29, Cuareim 1522).

2.1.2 Definición de modelo de datos

Un modelo de datos define las reglas que permiten estructurar los datos. Adicionalmente se necesitan *operaciones* sobre las estructuras de datos. La colección de datos generalmente se denomina *base de datos*. El término *base de datos* hace referencia a una instancia particular de la colección de datos y a una serie de instancias. Usualmente, una operación permitida en un modelo de datos transforma a la base de datos en una nueva base de datos. En la mayoría de los casos, ambas bases de datos tienen la misma estructura.

Definición: [Modelo de datos]

Un modelo de datos $M(G, O)$ consiste en un conjunto G de reglas generatrices y un conjunto O de operaciones.

El conjunto G expresa las *propiedades estáticas o estructurales* del modelo de datos, permitiendo definir las estructuras aceptadas para los datos del modelo. Los objetos permitidos se especifican usando reglas genéricas para definir categorías. Los objetos no permitidos son excluidos mediante la definición de *restricciones* sobre las categorías. Algunos modelos de datos particionan al conjunto G en dos partes: la especificación de estructuras G_s y la especificación de restricciones G_c . Los generadores G_s generan las

Este material es de uso exclusivo para los cursos impartido por Universidad de la Empresa y asociados



categorías y estructuras de un esquema y los generadores Gc generan las restricciones asociadas al esquema. Usualmente se denominan a las reglas generatrices como *lenguaje de definición de datos*.

Ejemplo:

En la definición del tipo de entidad *EMPLEADO*, el atributo *#Empleado* podría especificarse como un identificador (clave). Esto es una restricción que expresa que los empleados deben tener valores diferentes de *#Empleado* entre ellos.

Las *propiedades dinámicas* se expresan por el conjunto O de operaciones, que se corresponden al usualmente llamado *lenguaje de manipulación de datos*. El conjunto O define las acciones permitidas que pueden ejecutarse sobre una instancia de la base de datos, para lograr una nueva instancia de la misma. Si bien no todas las operaciones de O provocan un cambio en la base de datos, igualmente se las consideran "dinámicas" en el entendido de que causan un cambio en el estado de la base de datos.

Cada operación de O puede verse como una función que va un estado de la base de datos a otro estado de la base de datos, donde la nueva instancia está asociada al mismo esquema original. Esto es, una operación de O no modifica la estructura de los datos. Sin embargo, las operaciones de O se definen como funciones parciales, debido a que una operación de manipulación puede resultar en una base de datos que viola las restricciones asociadas a su esquema. Las implementaciones de sistemas de bases de datos manejan esta situación de distintas maneras, para garantizar que las operaciones de manipulación dan por resultado bases de datos consistentes (por ejemplo, mediante el manejo de transacciones).

2.2 ESTRUCTURAS

2.2.1 Abstracciones

Una de las formas de estructurar y visualizar datos es mediante el uso de la abstracción. La *abstracción* es la habilidad de ocultar los detalles y concentrarse en los detalles comunes generales a un conjunto de objetos. En el modelado de datos, se usa la abstracción para obtener categorías de datos. Una forma elemental de abstracción distingue entre el nivel de token y el nivel de tipo. Un *token* es un valor real o una instancia particular de un objeto. La abstracción se usa entonces para definir un *tipo* a partir de una clase de tokens similares.

En objetos de bases de datos, la abstracción se usa de dos maneras: generalización y agregación.

La *generalización* es una forma de abstracción que ve a un conjunto de tokens o a un conjunto de tipos, como un tipo genérico. La generalización a partir de tokens se denomina *clasificación* y la generalización a partir de tipos se denomina propiamente *generalización*. La *instanciación* es el proceso opuesto a la clasificación, y la *especialización* es el proceso opuesto a la generalización.

Ejemplo:

Este material es de uso exclusivo para los cursos impartidos por Universidad de la Empresa y asociados



Ver a un conjunto de tokens individuales de empleados como un tipo genérico EMPLEADO, se considera clasificación. Ver a dos tipos EMPLEADO y ESTUDIANTE como un tipo genérico PERSONA, se considera generalización.

Un token de empleado es una instanciación del tipo EMPLEADO, y el tipo EMPLEADO es una especialización del tipo PERSONA. Esto implica que cada empleado puede ser visto como una persona.

En general, un tipo generalizado puede tener todas las propiedades comunes de cada token o tipo constituyente. Es decir, todas las propiedades del tipo generalizado pueden ser heredadas por los tipos constituyentes. Sin embargo, la herencia de propiedades específicas puede prohibirse explícitamente. Adicionalmente, pueden definirse propiedades específicas a un tipo y que no son heredadas.

Ejemplo:

El concepto de que un empleado tiene un salario es heredado del tipo EMPLEADO por cada empleado individual. El concepto de que una persona tiene nombre, edad y domicilio es heredado por cada tipo participante del tipo PERSONA (e.g. EMPLEADO y ESTUDIANTE). El ejemplo de una propiedad específica para el tipo EMPLEADO y que no puede heredarse.

La *agregación* es una abstracción por la cual un objeto se construye a partir de sus objetos constituyentes. Por ejemplo, una persona puede estar caracterizada por su nombre, domicilio y edad. La agregación puede usarse a nivel de token o a nivel de tipo.

El tipo EMPLEADO puede construirse como agregación de los tipos NOMBRE, EDAD y DOMICILIO. Las propiedades de un tipo (tales como NOMBRE, EDAD y DOMICILIO) se denominan *propiedades intensionales*. La instancia de Juan Pérez como persona está construida por los tokens "11024" como número de empleado, "Juan Pérez" como nombre, "29" como edad y "Cuareim 1522" como domicilio. A estas propiedades de una instancia se las denominan *propiedades extensionales*.

La agregación implica que el objeto agregado tiene como parte de su estructura a sus objetos constituyentes. Las propiedades de cada objeto constituyente son adquiridas por cada objeto constituyente respectivo del objeto agregado.

2.2.2 Conjuntos, dominios y atributos

Un *conjunto* es una colección de objetos propiamente identificado y está caracterizado por una condición de pertenencia. Esta condición califica a un objeto como perteneciente o no, al conjunto. Algunos conjuntos tienen elementos más o menos homogéneos, por ejemplo, el conjunto de enteros entre 10 y 20, o el conjunto de strings alfanuméricos de largo 20. Estos conjuntos homogéneos se denominan *dominios*. Los dominios se usan en el modelado de datos como conjuntos de valores de los que ciertos objetos y sus propiedades, toman valores. Por ejemplo, los números de 6 dígitos forman un dominio del cual toman valores los salarios.



Un dominio con nombre, que representa objetos con cierta semántica, se denomina *atributo* y representa la intensión del dominio (e.g. Salario). La extensión del dominio, sus elementos, corresponde a los valores que puede tomar el atributo.

2.2.3 Entidades y relaciones

Una *relación n-aria* es un subconjunto de un producto cartesiano de n atributos. Los elementos de la relación son n -uplas (usualmente llamadas *tuplas*). En el modelado de datos, el concepto de relación define a un tipo. Una relación R es considerada como un tipo que define un conjunto de tuplas. Todas estas tuplas se definen sobre los mismos dominios y tienen propiedades similares.

Una interpretación semántica que puede aplicarse a una relación es hacer que cada tupla se corresponda con una *entidad* u objeto en particular. En este caso se dice que la relación representa a un *tipo entidad*. Si bien es difícil dar una definición rigurosa de entidad, se puede convenir que es algo que existe y tiene identidad propia. Cuando una relación representa a un tipo entidad, los dominios de la relación corresponden a los atributos del tipo entidad.

Es importante destacar que no hay distinción entre tipos entidad y atributos. A veces un atributo puede existir solamente como vinculado a un tipo entidad. En otro contexto puede ser un tipo entidad por derecho propio. En términos de abstracciones, un tipo entidad es una agregación de atributos.

Podemos interpretar a una relación como formada entre tipos entidad. Por ejemplo, la relación PAREJA puede interpretarse como un relacionamiento entre los tipos entidad HOMBRE y MUJER. Como los relacionamientos son también relaciones, el tipo relacionamiento se corresponde a una agregación de dos o más tipos entidad.

2.3 RESTRICCIONES

2.3.1 Introducción

Los atributos, tipos entidad y tipos relación, basados en conjuntos, pueden usarse para representar las estructuras de los datos. La especificación de estructura captura algunas de las propiedades de los datos. Adicionalmente se requieren otras propiedades que expresen restricciones sobre los valores de los datos y sobre cómo se relacionan los datos entre sí. Por ejemplo, una restricción del tipo "el salario de un jefe siempre es mayor que el salario de sus subalternos" no puede expresarse en términos de estructuras, mientras que una restricción del tipo "los salarios se expresan en dólares, con valores naturales menores a 10.000" sí puede expresarse únicamente en términos de estructuras.

Las restricciones lógicas sobre los datos se denominan *limitantes* o *restricciones de integridad*. Frecuentemente las denominaremos *restricciones*. Una restricción es una propiedad que para un conjunto, es o bien verdadera, o bien falsa. En el modelado de datos, las restricciones son útiles cuando son genéricas, es decir, cuando pueden definirse y aplicarse sobre un conjunto de objetos y no únicamente sobre una instancia particular de un objeto.



Las restricciones se requieren por razones semánticas y de integridad. Por razones semánticas, las restricciones permiten que los esquemas reflejen en forma más precisa una situación del mundo real. Por razones de integridad, las restricciones permiten a un sistema manejador de bases de datos limitar los valores posibles de forma tal que sólo se permitan los que las satisfacen.

Podemos identificar dos tipos básicos de restricciones: inherentes y explícitas. Las *restricciones inherentes* son parte integral de las estructuras del modelo de datos. Por ejemplo, los conjuntos imponen restricciones inherentes sobre los atributos, tipos entidad y tipos relación (e.g. elementos no duplicados y sin orden alguno). Este tipo de restricciones proveen facilidades bastante limitadas.

Algunos modelos de datos permiten la especificación de *restricciones explícitas* o *aserciones* en forma aparte a la especificación de estructura, mediante algún mecanismo de especificación de restricciones. Este mecanismo dispone de un lenguaje de especificación del conjunto de restricciones, un mecanismo de verificación que asegure que el conjunto de restricciones es consistente y algoritmos que garanticen que los datos de una base de datos cumplen con las restricciones.

Las restricciones explícitas pueden especificarse de dos maneras y usualmente se usan ambas. Una *especificación estática* expresa reglas que especifican cuáles son los estados de la base de datos que son permitidos. Usualmente se expresa mediante cálculo de predicados. Una *especificación dinámica* de las restricciones explícitas expresa cuáles transiciones de estado de la base de datos son permitidas. Las operaciones sobre la base de datos se definen de forma tal que solamente los estados legales resulten como consecuencia de una operación.

2.3.2 Dominios y atributos

Los atributos toman sus valores de los dominios asociados, que pueden estar definidos en términos de conjuntos. Los atributos heredan alguna restricciones de los conjuntos, a consecuencia de la condición de pertenencia. De esta forma, podemos definir un conjunto para restringir los valores que un atributo podrá tomar.

Ejemplo:

Podemos definir al conjunto de naturales de 20 a 65, que puede ser usado como base para un dominio N del cual el atributo Edad toma sus valores. Este dominio podría representar la restricción del mundo real que indique que los empleados puedan tener una edad entre 20 y 65. Sin embargo, usualmente vamos a requerir semántica adicional para el dominio. Por ejemplo, este mismo dominio podría ser la base para un atributo Velocidad. Si la restricción se limita únicamente a expresar que los valores de los atributos Edad y Velocidad son tomados del dominio N, no podríamos determinar si tiene sentido o no comparar una edad con una velocidad, o asignar a una edad un valor de velocidad.

En la práctica se da alguna semántica en base al uso de nombres de atributos adecuados (de hecho, sabemos que no tiene sentido comparar una edad con una velocidad). Sin embargo, solamente el uso de nombres de atributos adecuados no es suficiente. Por ejemplo, hay atributos que se nombran diferente y sin embargo son lo mismo desde el punto de vista semántico (e.g. Edad y Años). Una forma usual de



asociar semántica a un atributo es el concepto de *dominio interpretado*. El concepto de dominio interpretado se basa en la asociación de *unidades* a los dominios. De esta forma, podemos asociar la unidad "años" a Edad y "kilómetros por hora" a Velocidad. En este caso, dos atributos no son comparables a menos que las unidades de los atributos sean las mismas o puedan convertirse a una unidad común.

Los dominios interpretados pueden tener asociadas un conjunto de operaciones permitidas sobre sus elementos. Por ejemplo, podríamos disponer de una operación para incrementar en 1 a una edad, y no una operación para calcular el porcentaje de una edad (a menos que tenga algún sentido!).

2.4 EL MODELO ENTIDAD-RELACION

2.4.1 Introducción

El Modelo Entidad Relación (MER) es un modelo conceptual utilizado ampliamente para representar estructuras de información. No contiene lenguaje para representar operaciones de manipulación de información. En este modelo se utilizan entidades, conjuntos de entidades y relaciones entre éstos para estructurar la información.

2.4.2 Entidades y conjuntos de entidades

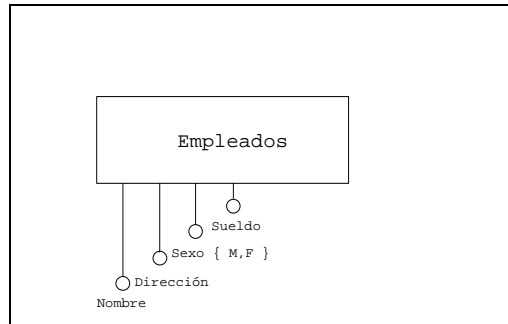
Una *entidad* es la representación abstracta de un objeto del mundo real que se desea modelar. Un grupo de entidades con características similares se denomina *conjunto de entidades*. En el MER representamos un conjunto de entidades dibujando un rectángulo con el nombre del conjunto escrito dentro de la figura.



Ejemplo 1. Conjuntos de Entidades

2.4.3 Atributos simples de entidades

A cada entidad en un conjunto de entidades le asociamos la información que deseamos "almacenar" sobre ella. Estas *asociaciones* son funciones (en sentido matemático) que van de una entidad del conjunto de entidades hacia un valor de un *conjunto de valores*. A estas asociaciones se las denomina *atributos del conjunto de entidades*.



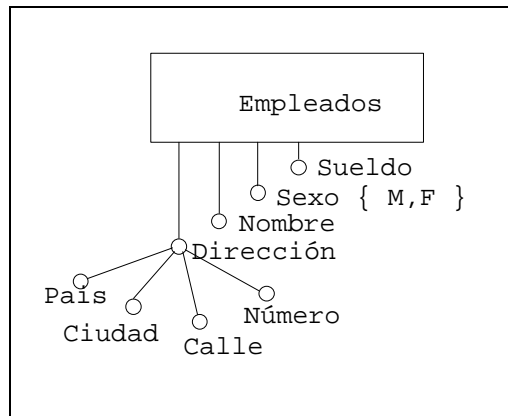
Ejemplo 2. Atributos Simples.

Asociaciones:

Nombre: Empleados --> Conjunto de nombres
Domicilio: Empleados --> Conjunto de domicilios
Sexo: Empleados --> {M,F}
Sueldo: Empleados --> Reales

2.4.4 Atributos complejos

Frecuentemente vamos a representar atributos que pueden estar compuestos de *subatributos*. Esta situación establece una estructura arborescente. Por ejemplo, el domicilio de los funcionarios podría formarse con el país, ciudad y calle donde viven.

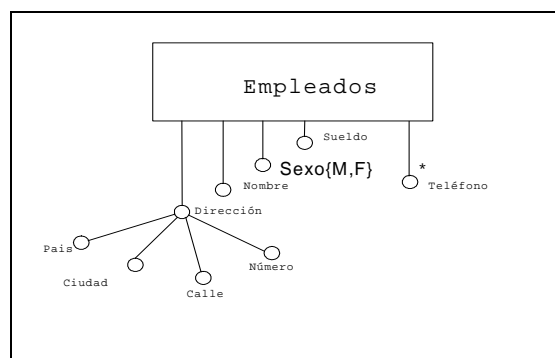


Ejemplo 3. Atributos Complejos.

Con esta representación podemos referirnos al atributo Domicilio de un Empleado y el valor de este atributo será tomado como la concatenación de tres valores: uno de País, uno de Ciudad y otro de Calle. En este caso la función Domicilio retorna una terna (un valor del producto cartesiano de tres conjuntos de valores: el conjunto de países, el conjunto de ciudades y el conjunto de calles).

2.4.5 Atributos monovalorados y multivalorados

Cuando un atributo asume un único valor para cada entidad, se denomina *atributo monovalorado*, como es el caso del atributo Nombre. En muchos casos vamos a permitir funciones multivalentes: en este caso un atributo asume un conjunto de valores para cada entidad y se denomina *atributo multivalorado*. Lo representaremos gráficamente marcando con un asterisco. Un ejemplo típico es el conjunto de teléfonos asociados a un empleado, bajo la suposición de que el empleado puede ser localizado en más de un teléfono posible.

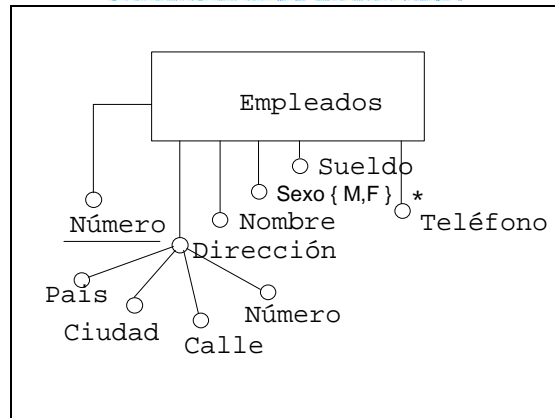


Ejemplo 4. Atributos Multivalorados.

Como observación: permitiremos que los atributos (mono y multivalorados) tengan un *valor vacío*, que se denomina *valor nulo* o *NULL* en representaciones operacionales y en implementaciones.

2.4.6 Atributos determinantes

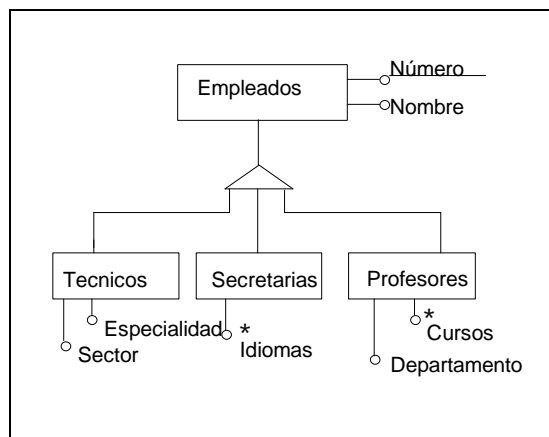
Una restricción de integridad muy común en conjuntos de entidades es el hecho de que un atributo sea una función biunívoca. A estos atributos los denominaremos *atributos determinantes* y los representaremos gráficamente subrayando el atributo. Por ejemplo, podemos representar que cada Empleado tiene un único número de empleado (no hay dos empleados con el mismo número). En modelos "más operacionales" se denominan a los atributos determinantes como *clave*.



Ejemplo 5. Atributo Determinante.

2.4.7 Categorización de entidades

En muchos casos prácticos un conjunto de entidades representa a elementos del mundo real que se subdividen en *categorías* con atributos en parte distintos. Veamos el siguiente caso:



Ejemplo 6. Categorizaciones.

En la figura anterior tenemos que los empleados se subdividen en técnicos, secretarias y profesores. Todos ellos tienen los atributos Número (determinante) y Nombre. Cada categoría tiene atributos específicos, por ejemplo, un profesor tiene una serie de cursos que dicta y un departamento para el cual trabaja. Por otro lado, una secretaria conoce un conjunto de idiomas y un técnico trabaja en cierto sector y tiene alguna especialidad profesional.

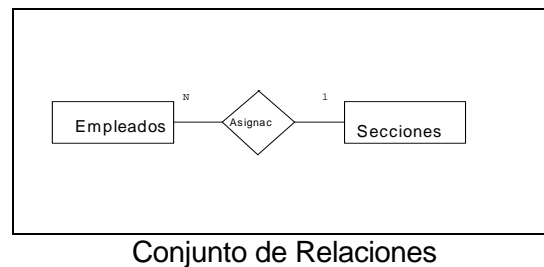
2.4.8 Relaciones entre conjuntos de entidades

Necesitamos representar en nuestro modelo las asociaciones entre las entidades de un conjunto de entidades, con las entidades de otros conjuntos de entidades. Esta estructura abstracta se denomina *relación*. Una relación binaria es una pareja ordenada

Este material es de uso exclusivo para los cursos impartido por Universidad de la Empresa y asociados

(e1, e2) donde e1 y e2 son respectivamente entidades de dos conjuntos de entidades E1 y E2.

Como ejemplo, supongamos los conjuntos de entidades Empleados y Secciones. Podemos construir parejas (e, s) con entidades de ambos conjuntos, para representar dónde ha sido asignado a trabajar cada empleado. El *conjunto de relaciones* (en este caso, es un conjunto de parejas) expresa las asignaciones de los empleados a sus respectivas secciones de trabajo. Representaremos a este conjunto de relaciones dibujando un rombo con el nombre del conjunto escrito dentro.



El diagrama anterior representa el hecho de que los empleados se asocian a las secciones a través de un conjunto de relaciones Asignac. Este conjunto se compone de parejas puramente estructurales. La única semántica de tales parejas estará dada por los valores contenidos en ella.

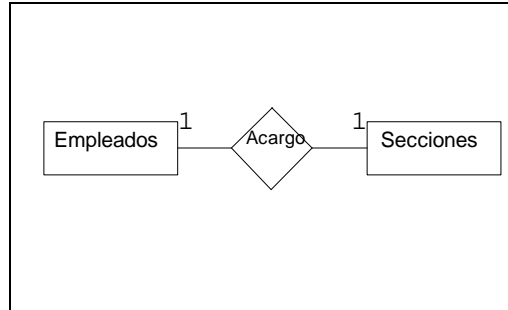
A cada conjunto de relaciones se le asocian *papeles* o *roles*. Por ejemplo, tendremos los roles "empleado asignado a" para cada entidad de Empleados y "sección emplea a" para cada entidad de Secciones. Los elementos de Asignac son parejas ordenadas (x,y) que pueden leerse como "x es un empleado asignado a y" o como "y es una sección que emplea a x".

Si consideramos en forma genérica a dos conjuntos de entidades E1 y E2, formalmente el conjunto de relaciones R se define como un subconjunto del producto cartesiano E1 x E2.

2.4.9 Clases de relaciones

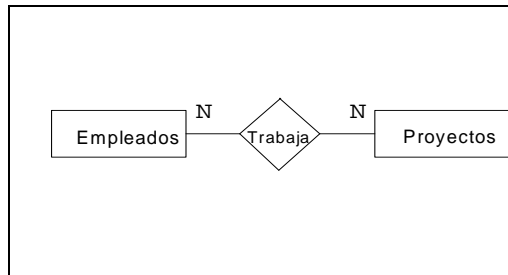
En el dibujo anterior observamos una "N" y un "1" en las salidas del conjunto de relaciones. Ellos determinan la clase de relación: decimos que Asignac es una relación de clase *N a 1* o *N:1* o de *muchos a uno*. Intuitivamente queremos representar que cada empleado está asignado a una única sección, o que cada sección emplea a varios empleados. Formalmente, diremos que una relación R entre dos conjuntos de entidades A y B es de N:1 si cuando (ak, bm) y (ak, bn) pertenecen a R entonces bm = bn.

Una relación de clase *1 a 1* o *1:1* o de *uno a uno* puede verse en la siguiente figura, en donde expresamos que cada sección tiene un único jefe a cargo:

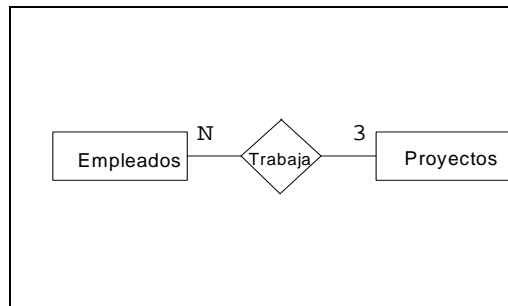


Formalmente, diremos que una relación R entre dos conjuntos de entidades A y B es de clase 1:1 si cuando (a_k, b_m) , (a_k, b_n) y (a_r, b_m) pertenecen a R entonces $a_k = a_r$ y $b_m = b_n$. Observemos que una relación de clase 1:1 es un caso particular de una de clase 1:N.

Una relación de clase N a N o $N:N$ o de *muchos a muchos* puede verse en la siguiente figura, en donde expresamos que cada empleado puede estar trabajando en varios proyectos a la vez y que cada proyecto esta formado por varios empleados.



Permitiremos colocar una constante numérica en vez de la "N" cuando deseemos expresar que hay ciertas restricciones de cardinalidad conocidas de antemano. Por ejemplo, si sabemos que un empleado no puede estar trabajando en más de 3 proyectos a la vez. A esta clase de relación la llamamos $c:N$ (donde c es algún natural).

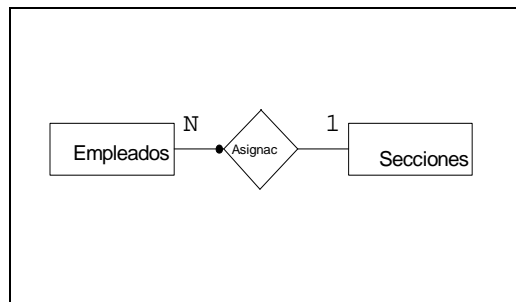


Formalmente, diremos que una relación R entre dos conjuntos de entidades A y B es de clase $c:N$ si cuando (a_1, b) , (a_2, b) , ..., (a_n, b) pertenecen a R y $a_i \neq a_j$ para $i, j \leq n$ entonces $n \leq c$.

2.4.10 Relaciones parciales y totales

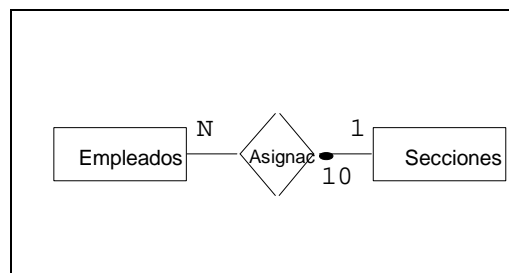
Una restricción muy común e importante en el modelado de casos prácticos impone que todas las entidades de un conjunto de entidades E aparezcan obligatoriamente en un conjunto de relaciones R. En estos casos decimos que la relación R es *total* en E. Cuando una relación "no es total", decimos que es *parcial*.

Como ejemplo, supongamos que deseamos expresar que todo empleado obligatoriamente está asignado a una sección (es decir, no hay empleados trabajando por su cuenta fuera de las secciones). En esta situación decimos que la relación Asignac "es total" en Empleados. Gráficamente colocamos un círculo en la arista que sale hacia Empleados:



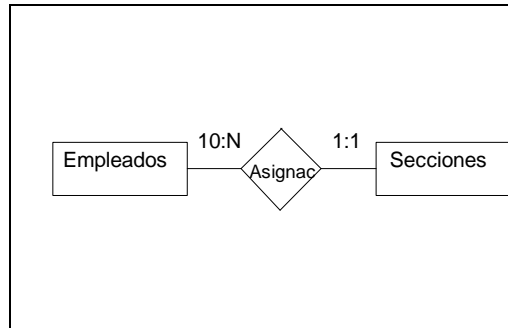
Relaciones Totales.

Es posible también expresar *parcialidad mínima* en las relaciones parciales, tal como se ve en la siguiente figura:



Parcialidad mínima.

En este caso expresamos que cualquier sección ha de tener 10 o más empleados asignados a ella, y que cada empleado está obligatoriamente asignado a una única sección. Algunos autores prefieren anotar una pareja que marca las cardinalidad mínima y máxima:

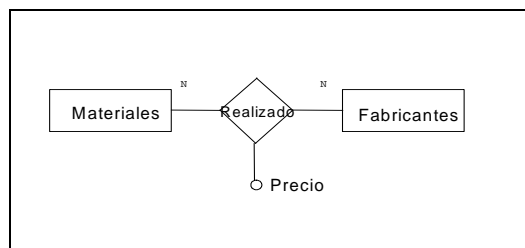


Cardinalidad mínima y máxima.

Esta notación expresa que "una sección tiene como mínimo 10 empleados y máximo N" y que "un empleado está asignado a 1 sección como mínimo y a 1 sección como máximo" (es decir, una única sección).

2.4.11 Atributos de relaciones

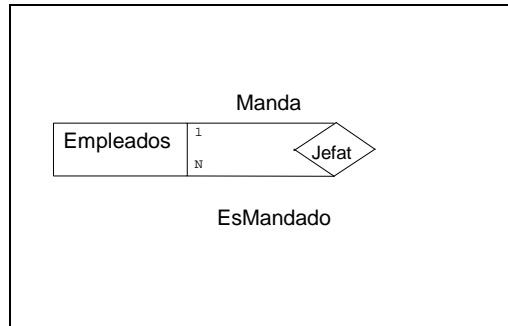
En muchos casos prácticos tenemos atributos que no dependen de una única entidad sino de la relación existente entre conjuntos de entidades. Supongamos ahora los conjuntos de entidades Materiales y Fabricantes, relacionados por Ventas. En este caso, el precio de venta de un material depende de dos cosas: del material y del fabricante del mismo:



Atributos de relaciones.

2.4.12 Auto relaciones

Sea R un conjunto de relaciones que vincula elementos de un conjunto de entidades E con entidades de ese mismo conjunto E. Se denomina a R, *auto relación*. En la siguiente figura expresamos una auto relación entre empleados que establece qué empleado manda sobre cuál.

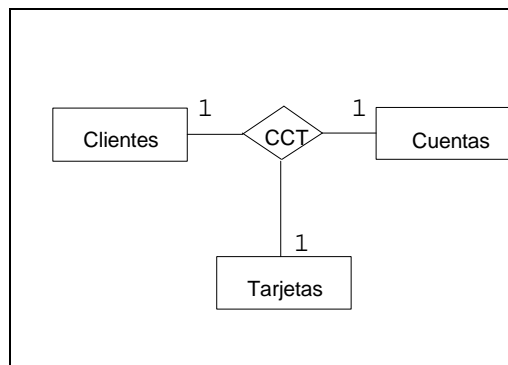


Auto relaciones.

En la figura expresamos que cualquier empleado es mandado por un único empleado y que un empleado puede mandar a varios empleados. Obsérvese el uso de roles para aclarar la situación.

2.4.13 Agregaciones

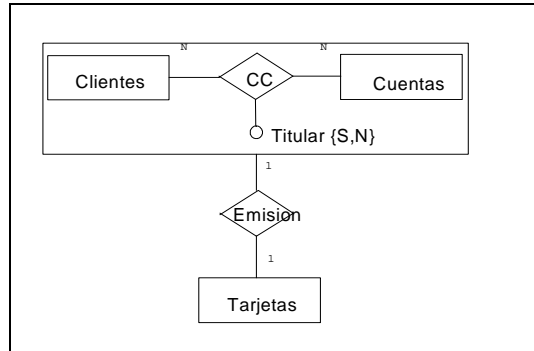
Introduzcamos el siguiente caso práctico: supongamos un banco que trabaja con clientes que pueden tener varias cuentas corrientes. A pedido de los clientes y bajo ciertos requisitos, el banco les otorga tarjetas magnéticas para ser usadas en sus cajeros automáticos. Una primera aproximación puede ser:



Primera versión.

Este esquema establece que se emite una tarjeta por cada cuenta que tenga cada cliente. Sin embargo, la situación que deseamos plantear es que se emiten tarjetas solamente a pedido de los clientes y cuando sus cuentas cumplan ciertos requisitos (por ejemplo, antigüedad o monto disponible en la cuenta). Es decir, cada tarjeta debe vincularse a una pareja (cliente,cuenta).

A tales efectos deberíamos considerar a las parejas (cliente,cuenta) como un conjunto de entidades que se vinculan con entidades del conjunto de tarjetas. Para ello encerramos a los conjuntos Clientes, Cuentas y la relación entre ellos, en un nuevo rectángulo que se va a comportar como un nuevo conjunto de entidades. A esto lo denominamos *agregación*. Los elementos de esta agregación se relacionan con las tarjetas. Gráficamente:

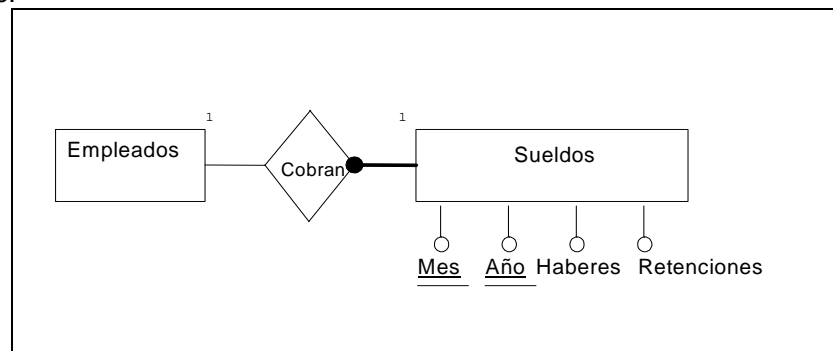


Versión mejorada.

Hemos colocado algunos atributos para ejemplificar. El atributo Titular expresa si el cliente es titular de la cuenta, ya que tenemos que una cuenta puede asociarse a varios clientes. Observemos que algunos pares (cliente,cuenta), pero no todos, tienen a lo máximo una tarjeta. Cada tarjeta es siempre emitida para una pareja (cliente,cuenta).

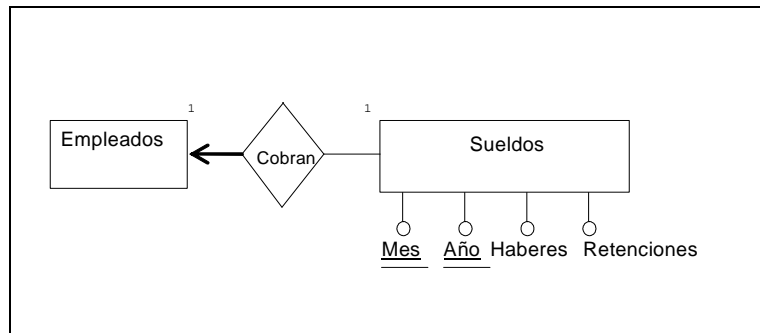
2.4.13 Entidades Débiles

Supongamos, por ejemplo, que un empleado cobra un sueldo mensual y se quiere registrar cada uno de los sueldos cobrados, incluyendo el mes y el año, los haberes y las retenciones y que no se puede ingresar un sueldo si no existe un empleado asociado. Si lo representamos como hemos visto hasta ahora, tendríamos el siguiente modelo:

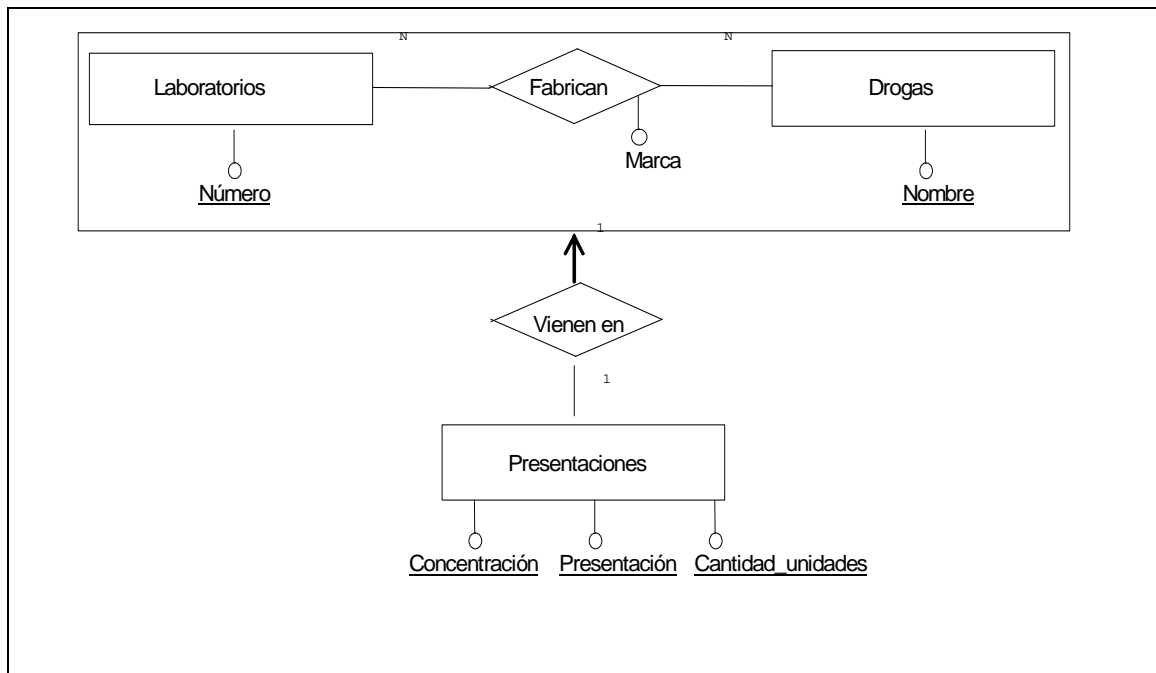


Sin embargo, un sueldo por sí solo no puede ser identificado utilizando solamente mes y año, sino que requerimos además el empleado para poder saber de qué sueldo se trata.

Una entidad débil es aquella que no puede existir si no existe una entidad relacionada. Es decir que una entidad débil es una entidad que depende de otra para existir y ser identificada. Se representa mediante una flecha del lado de la entidad que no es débil y en tal caso, se puede omitir el punto relativo a relación total, ya que queda implícito por la presencia de la flecha.

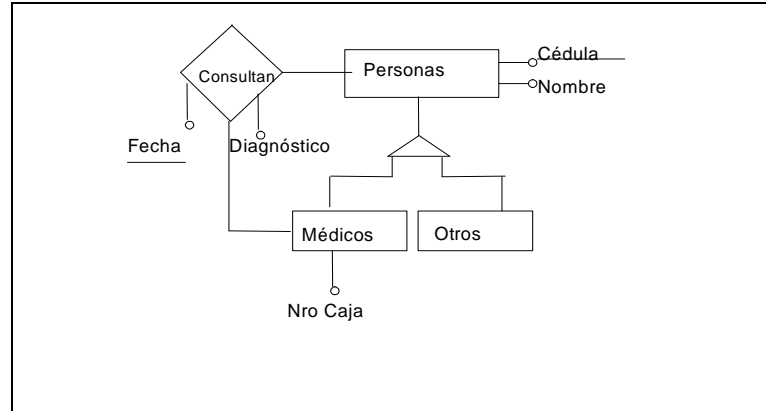


Una entidad también puede ser débil respecto a una relación, como se muestra en el siguiente ejemplo:



2.4.14 Determinantes en Relaciones

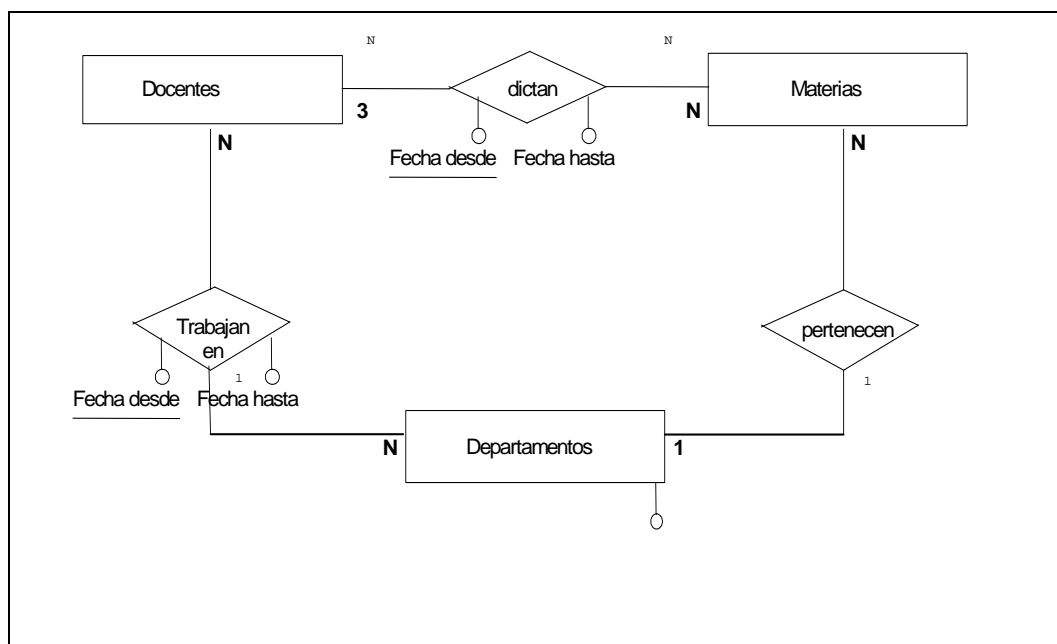
En el siguiente ejemplo se muestra un determinante correspondiente a una relación. Existen personas que pueden ser médicos u otros, y que realizan consultas a médicos. Ahora bien, una persona puede consultar a un médico más de una vez, registrando la fecha y el diagnóstico realizado. Para poder diferenciar e identificar las diferentes consultas, además de la Persona y el Médico que participan de la consulta, se requiere incluir también a la fecha en el determinante, como se muestra a continuación:



2.4.15 Restricciones no estructurales

Las restricciones no estructurales son restricciones adicionales que no se pueden expresar en el modelo. Se pueden expresar en lenguaje natural o preferentemente en lenguaje formal. Se pueden expresar como notas dentro del dibujo o en una lista separada, vinculada al modelo.

En el siguiente ejemplo, un docente dicta varias materias, y cada materia puede ser dictada por hasta 3 docentes. Cada materia pertenece a un departamento, el cual puede tener varias materias asignadas. A su vez, los docentes trabajan en un departamento en un cierto período. El siguiente modelo expresa dicha situación:





Queremos agregar la siguiente restricción: “Un docente no puede dictar una materia de un departamento en un período que no esté incluido en un período en el que haya trabajado en dicho departamento”

Dicha restricción no puede ser expresada con la notación del modelo, por lo que se requiere agregar una restricción no estructural. Formalmente:

$$\begin{aligned} & \forall d \in \text{DOCENTES} \wedge m \in \text{MATERIAS} / r \in \text{DICTAN} \wedge r.\text{docente} = \\ & d \wedge r.\text{materia} = m \\ & \Rightarrow t \in \text{TRABAJAN} / t.\text{docente} = d \wedge p \in \text{PERTENECEN} / \\ & p.\text{departamento} = t.\text{departamento} \wedge p.\text{materia} = m \wedge t.\text{fecha_desde} \\ & \leq r.\text{fecha_desde} \wedge t.\text{fecha_hasta} \geq r.\text{fecha_hasta} \end{aligned}$$

Bibliografía.

[Setzer 1986]. V.W. Setzer. Brasil 1986. *Projeto lógico e projeto físico de bancos de dados*, V Escola de Computação.

[Tsichritzis 1982]. D. Tsichritzis y F.H. Lochovsky. 1982. *Data Models*, Prentice-Hall.