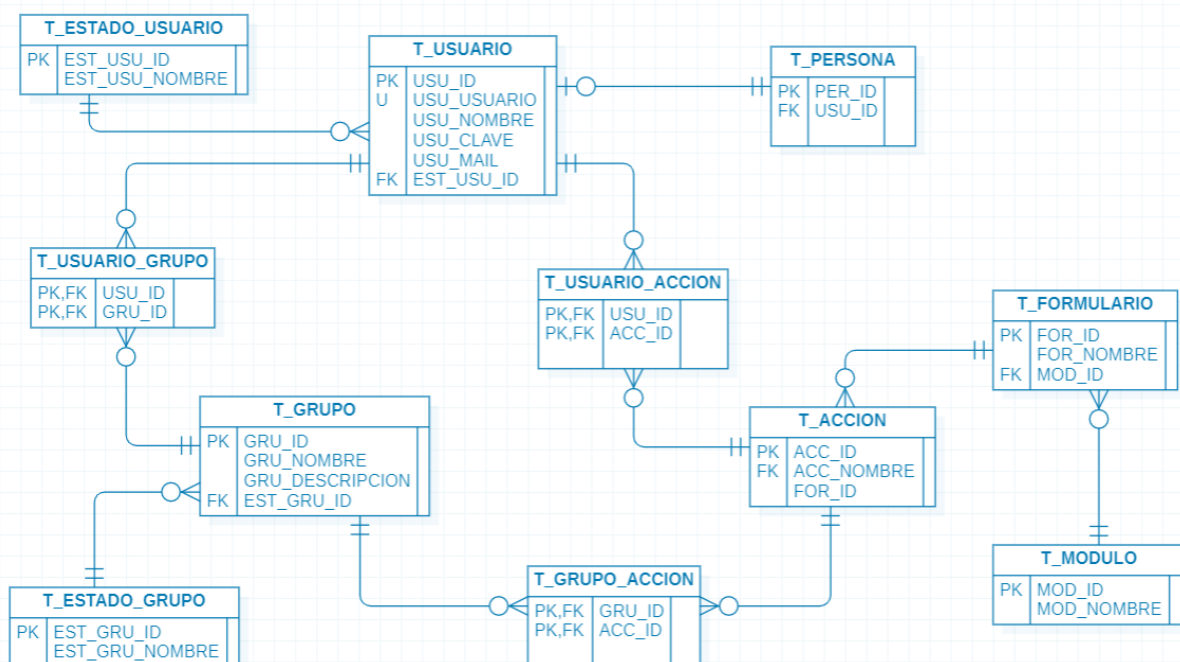


Modelo de Datos Relacional



Contenido

Conceptos Generales	4
Memoria Esencial.....	4
Dato	4
Información	4
Base de datos	4
Redundancia de datos	5
Inconsistencia en los datos	5
Modelo de Datos	5
Objetivos de un Modelo de Datos	5
¿Qué relación existe entre estos objetivos?	5
Construcción del Modelo de Datos.....	6
Diagrama de Entidad Relación (DER)	6
Entidad	7
Relación	9
Entidad Asociativa	11
Subtipo	13
Subtipificación	16
Supertipo	17
Como detectar entidades y relaciones	18
Cómo refinar el DER.....	20
Cómo realizar el Diccionario de Datos de Entidades que completa el DER	20
Cómo "mapear" el DER al gráfico que forma parte de un Mapa de Información Preliminar	22
Vinculaciones	22
Vinculación 1 a 1	23
Vinculación 1 a N (muchos)	23
Vinculación N (muchos) a 1	24
Vinculación N a M (muchos a muchos)	25
Optimización del Mapa preliminar	29
Algunos conceptos previos a la Normalización.....	29
Clave candidata.....	29
Clave primaria	29
Clave alternativa.....	29
Dependencia Funcional	29
Dependencias completas y no completas	30

Carácter unívoco	31
Transitividad	31
Normalización	32
Proceso de normalización	32
Primera forma normal (1FN)	33
Segunda forma normal (2FN)	33
Tercera forma normal (3FN)	34
Cómo evolucionó la entidad de nuestro caso	35
Porque conviene normalizar	36
Problemas con la primera forma normal	37
Problemas con la segunda forma normal	37
Conclusiones.....	38
Mapa de Información Canónico (o Normalizado)	38
Buenas Descomposiciones.....	39
¿Cómo reconocer cuándo dos entidades son independientes?	40
Visiones para validar el Mapa de Información Canónico o Normalizado	41
Resumen.....	42
Pasos propuestos para el modelado de datos:.....	42

Conceptos Generales

Antes de comenzar con el modelado de una base de datos relacional, debemos definir algunos conceptos generales que permiten aclarar un poco el panorama.

Memoria Esencial

La memoria esencial de un sistema son todos los datos que el sistema debe almacenar o guardar para poder producir las respuestas a todos los requerimientos que lo afectan.

Un sistema posee dos fuentes de información:

- el ambiente: que provee al sistema parte de lo necesario para producir sus respuestas.
- la propia memoria del sistema, alcanzada por el término de tecnología perfecta. Esto significa que el sistema podrá acceder a su memoria de cualquier forma y además, desde el punto de vista esencial, será ilimitada.

Pero ¿por qué un sistema necesitará recordar datos? Porque el ambiente no es perfecto (no siempre se acuerda de todo), y/o no es confiable. Es decir, el concepto de tecnología perfecta no se extiende al ambiente que rodea el sistema. Por ejemplo:

- en un sistema de Ventas no podemos pretender que cada vez que el sistema deba generar una factura, el ambiente tenga que informarle todos los datos del cliente y todos los datos del producto y todos los datos de los impuestos a aplicar para que el sistema se limite a actuar como una simple calculadora.

- en un sistema de Sueldos y Jornales no podemos pretender que cada vez que se tengan que liquidar los sueldos de los empleados sea necesario ingresar la antigüedad y la categoría. Alguien podría falsear estos datos para cobrar un poco más. En este caso, el ambiente no sería confiable.

Dato

Desde el punto de vista de base de datos, podemos decir que un dato es algo que refleja el valor de una característica de un objeto real, sea concreto o abstracto. Debe cumplir algunas condiciones, por ejemplo, debe permanecer en el tiempo. En ese sentido, estrictamente hablando, una edad no es un dato, ya que varía con el tiempo. El *dato* sería la *fecha de nacimiento*, y la edad se calcula a partir de ese dato y de la fecha actual.

Un dato tiene un dominio de valores posibles, que es el conjunto de valores que puede tomar.

Información

Información es el conjunto de datos procesados en forma significativa, ordenados y con una secuencia lógica sobre algún suceso o hecho de importancia. Con valor real para la toma de decisiones, a medida que tenemos más información, más fácil nos resulta tomar decisiones correctas. Esa es la función de la información: disminuir la incertidumbre o aumentar el conocimiento, incrementando además la probabilidad de éxito.



Base de datos

Podemos considerar que es un conjunto de datos de varios tipos, organizados e interrelacionados. Estos datos deben estar libres de redundancias innecesarias y ser independientes de los programas que los usan.

Redundancia de datos

Decimos que hay redundancia de datos cuando la misma información es almacenada varias veces en la misma base de datos. Esto es siempre algo a evitar, la redundancia dificulta la tarea de modificación de datos, y es el motivo más frecuente de inconsistencia de datos. Además requiere un mayor espacio de almacenamiento, que influye en mayor coste y mayor tiempo de acceso a los datos.

Inconsistencia en los datos

Sólo se produce cuando existe redundancia de datos. La inconsistencia consiste en que no todas las copias redundantes contienen la misma información. Así, existen diferentes modos de obtener la misma información, y esas formas pueden conducir a datos almacenados en distintos sitios. El problema surge al modificar esa información, si sólo cambiamos esos valores en algunos de los lugares en que se guardan, las consultas que hagamos más tarde podrán dar como resultado respuestas inconsistentes (es decir, diferentes). Puede darse el caso de que dos aplicaciones diferentes proporcionen resultados distintos para el mismo dato.

Modelo de Datos

El modelo de datos de un sistema es la explicitación de la forma en que la memoria esencial del sistema está estructurada y de las relaciones existentes entre sus componentes, teniendo siempre presente la realidad sobre la cual el sistema opera.

El modelo de datos debe construirse con profundidad, teniendo como objetivo modelar la estructura de los datos de manera tal que garantice el funcionamiento del sistema a lo largo del tiempo y facilite la tarea de diseño o ampliación de la estructura si fuese necesario.

Objetivos de un Modelo de Datos

Es deseable que todo modelo de datos cumpla los siguientes objetivos:

1. **Estructura mínima:** que va mucho más allá del potencial ahorro de espacio de almacenamiento, sino que está íntimamente ligado a los conceptos de no redundancia e inconsistencia, y en consecuencia de integridad.
2. **Que asegure la no pérdida de información:** es decir, que le permita al sistema responder a todos los requerimientos respetando las restricciones existentes.
3. **Que facilite los procesos de mantenimiento (ABM):** el mantenimiento en tiempo y forma de los datos es importantísimo y ABMs simplificados se traducen en software simplificado, y ello en calidad del sistema.
4. **Máxima supervivencia:** con la máxima capacidad para soportar cambios y nuevos requerimientos.

¿Qué relación existe entre estos objetivos?

El concepto de no redundancia hace que se faciliten los procesos de mantenimiento o ABMs, ya que de no cumplirse esto, se podría llegar a tener un mismo conjunto de datos distribuidos en distintos lugares del modelo; lo que obligaría, al hacer su mantenimiento, a recordar que es necesario actualizarlos en dos o más lugares diferentes.

El concepto de integridad implica la completitud del Modelo en lo referente a la relación de los datos.

Se deberá buscar una estructura mínima óptima que a su vez asegure la no pérdida de información, validando que el sistema pueda seguir produciendo las respuestas, respetando las visiones de contexto, y en un todo con los objetivos planteados.

Construcción del Modelo de Datos

No existe una “receta”, sino herramientas y métodos para construirlo y formas de validarlo. Podemos encarar la construcción del mismo, dividiéndolo en etapas:

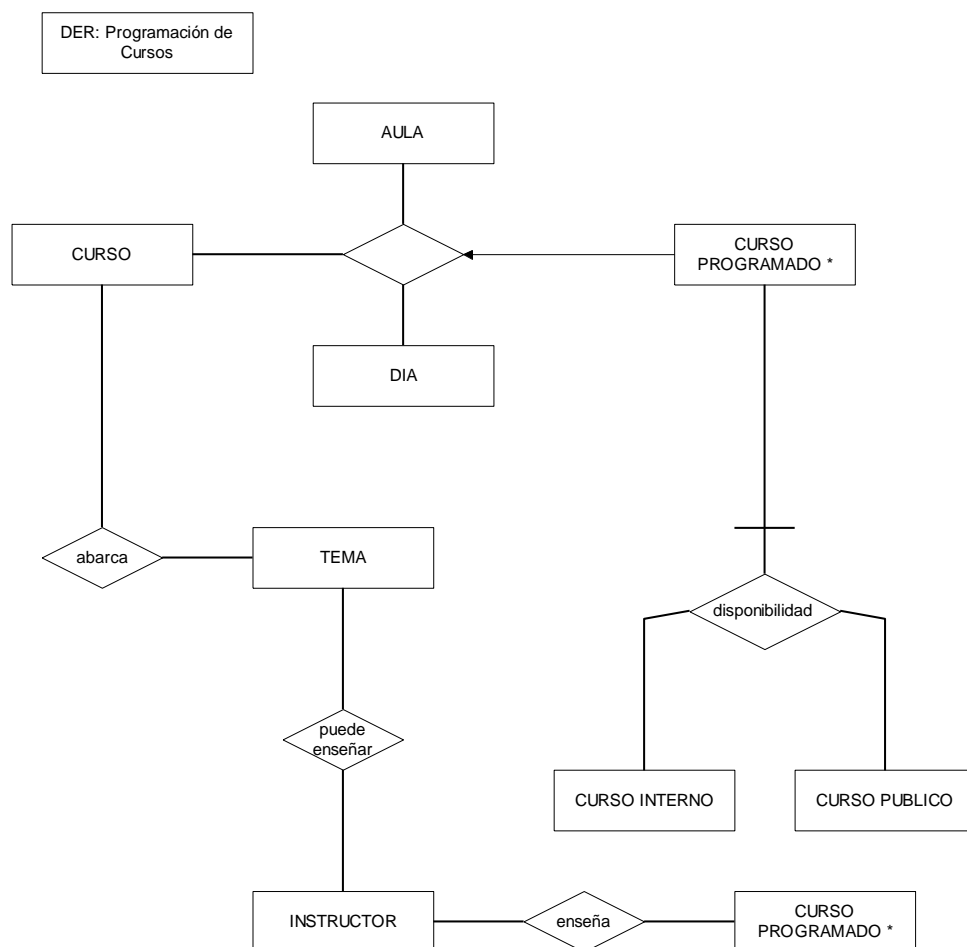
1. Construir y refinar el Diagrama de Entidad Relación (DER), en el cual explicitamos los componentes y las relaciones entre ellos.
2. Mapear el DER para lograr el Mapa de Información Preliminar.
3. Optimizar el mapa anterior, tratamos de satisfacer los objetivos a cumplir; afianzamos los componentes ; construimos un **Mapa de Información Canónico o Normalizado** y lo validamos.

Diagrama de Entidad Relación (DER)

Un diagrama de entidad relación es una herramienta gráfica para identificar los objetos o elementos que le interesan al sistema, que componen la memoria esencial del mismo y determinar cómo se relacionan.

Un DER muestra un conjunto de objetos del mundo real que son significativos para el sistema y sobre los cuales necesita almacenar información.

El Modelo de Datos usa el DER para mostrar las entidades y relaciones de las cuales el sistema es responsable.



Entidad

Una entidad es un objeto componente de la memoria esencial sobre la cual el sistema necesita recordar o almacenar información. Se caracteriza porque:

- Se describe por uno o más atributos o elementos de datos.
- Posee una o más ocurrencias.
- Cada ocurrencia posee uno o más atributos que la identifican de forma única.

Se representa con un rectángulo y un nombre:

NOMBRE

Entidad: ejemplo

INSTRUCTOR

Una entidad es una clase de objeto del mundo real cuyo rol o función dentro del sistema está bien definido. Puede corresponder a un objeto tangible, intangible o a conceptos abstractos. El sistema usará información de una entidad, interactuando con ocurrencias de la misma. Toda entidad tiene un nombre único que debería reflejar el rol que posee dentro del sistema.

Una entidad representa en su totalidad a un conjunto de posibles ocurrencias. Por ejemplo, la entidad *Curso* podría tener muchas ocurrencias. Tres de estas ocurrencias podría ser "Introducción al Cálculo Numérico", "Virus Informáticos - cómo combatirlos" y "Musicoterapia - nuevos avances".

El sistema identificará ocurrencias individuales de una entidad y la información almacenada sobre ella. Esta información puede referirse a las relaciones en las que participa la entidad o a los atributos que posee. Cada ocurrencia de una entidad debe ser distinta de las demás, pero debe cumplir el mismo rol que las demás ocurrencias de la entidad.

Supongamos que existe una entidad en nuestro DER llamada *Instructor*, sobre la cual podemos decir que: "...Existe una clase de personas del mundo real que poseen ciertas propiedades en común. Instructor es el nombre genérico con el cual se llama a cualquiera de ellas. Existen muchas ocurrencias de Instructor, cada una unívocamente identificada. Ciertos aspectos de Instructor (tales como sus atributos o su participación en relaciones) son de importancia para el sistema...".

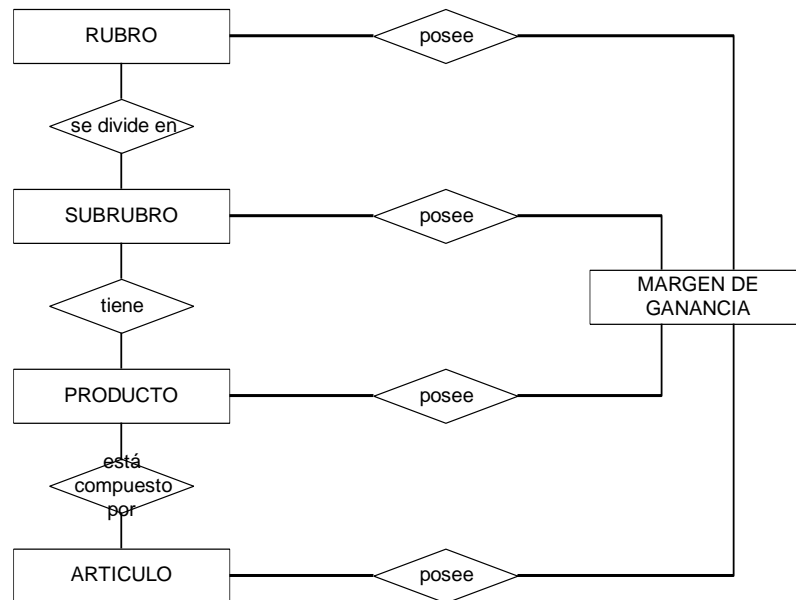
Algunas entidades representarán objetos tangibles físicos que juegan roles particulares en su interacción con el sistema. En términos generales son fáciles de identificar y presentan poca dificultad. Por ejemplo: para una empresa que dicta cursos para empresas, las entidades *Instructor* y *Aula* son importantes.

Otras entidades tienen una naturaleza más abstracta. Incluyen entidades que representan períodos de tiempo como *Día* y conceptos abstractos como *Curso*.

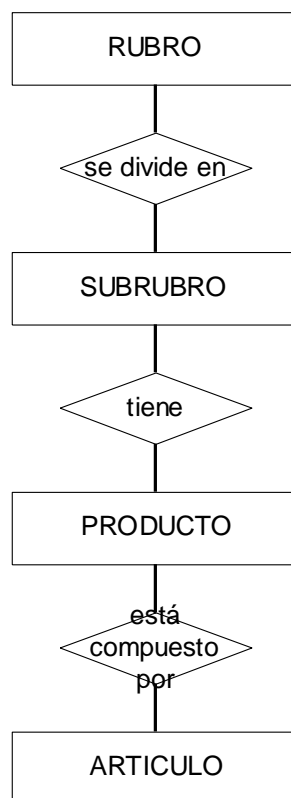
Como dijimos anteriormente, si una entidad *Instructor* forma parte de nuestro DER, entonces, existe un conjunto de cosas del mundo real, cada una de las cuales es una ocurrencia de *Instructor*. Cada una de estas ocurrencias debería poder referenciarse (por ejemplo, señalándola) y afirmar: Esto es un Instructor. Esta afirmación debe ser verificable; es decir: debe poder determinarse si es verdadera o falsa. Si bien no es posible "apuntar" a entidades abstractas como *Curso*, debería ser posible referenciar ocurrencias individuales y declarar que son ocurrencias de esta entidad.

Si una potencial entidad no satisface este criterio, entonces no es una entidad.

Por ejemplo: para un negocio que se dedica a la venta de artículos de ferretería, podríamos haber modelado lo siguiente (fragmento del DER del subsistema de ventas)



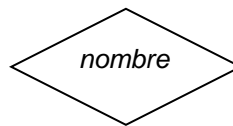
Sin embargo, una revisión más detallada de este fragmento del DER nos permite concluir que *Margen de Ganancia* no posee ocurrencias a las cuales podamos referenciar o apuntar, o bien describirlas con atributos. La entidad deberá transformarse en atributo de *Rubro*, *Subrubro*, *Producto* y *Artículo*, y el DER quedaría :



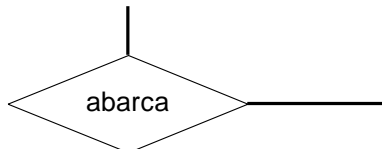
Relación

Una relación representa una posible asociación que puede darse entre ocurrencia de entidades. Cada ocurrencia de una relación corresponde a ocurrencias específicas de aquellas entidades que están relacionadas.

Se representa con un rombo:

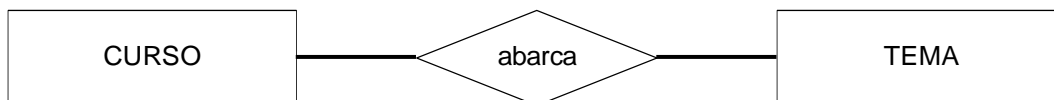


Relación: ejemplo



Una relación representa una posible asociación que puede darse entre ocurrencia de entidades. Cada ocurrencia de una relación corresponde a ocurrencias específicas de aquellas entidades que están relacionadas.

Una relación puede considerarse como un patrón al cual se le puede adicionar referencias a entidades específicas para obtener hechos específicos acerca del mundo real. Por ejemplo:



actúa como un template o patrón en el cual ocurrencias de *Curso* y *Tema* pueden ingresarse para obtener hechos específicos:

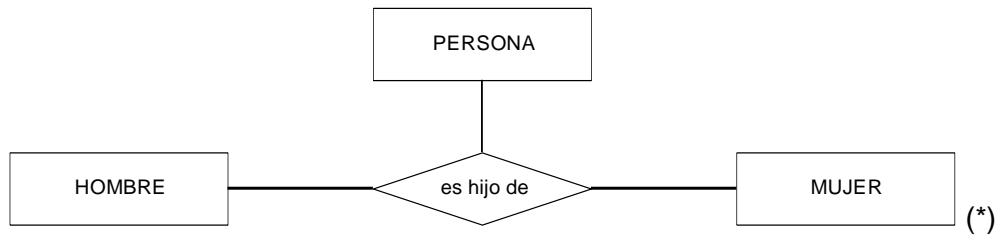
Introducción al Ruso	abarca	el alfabeto Cirílico
Master en navegación	abarca	medición angular
Astronomía esférica	abarca	medición angular

Cada ocurrencia de una relación corresponde a una asociación de cero o más ocurrencias de cada una de las entidades que participan en la relación.

Cada ocurrencia de una relación muestra que un evento ocurrió y que éste involucró a las ocurrencias específicas de las entidades que participan en la relación. Algunas relaciones sólo recuerdan la ocurrencia de un evento (Ej.: *Persona* se casó con *Persona*); se dice que son relaciones tipo “trace”, ya que registran el evento. Otras, en cambio reflejan una relación de continuidad entre entidades (Ej.: *Persona* está casada con *Persona*); se dice que son asociaciones continuas. La heurística dice que las relaciones tipo “trace” tienen una alta probabilidad de tener atributos y transformarse entonces, en entidades asociativas.

Relaciones de orden superior:

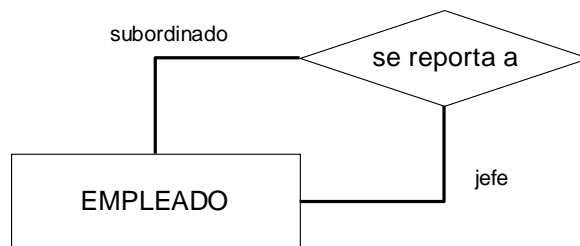
Una relación que involucre a más de dos ocurrencias de entidades se dice que es de orden superior. Una relación de 3º orden aparece en el DER con tres líneas que salen del diamante. Por ejemplo:



puede indicarse: *Persona* es hijo de *Hombre* y *Mujer*

Relaciones recursivas:

Muchas veces una relación involucra ocurrencias de una misma entidad, por ejemplo:



Cuando una entidad se repite en una relación, se dice que la relación es recursiva.

Para indicar en el ejemplo cuál de los empleados se reporta a quién, indicamos:

Empleado(subordinado) se reporta a *Empleado*(jefe)

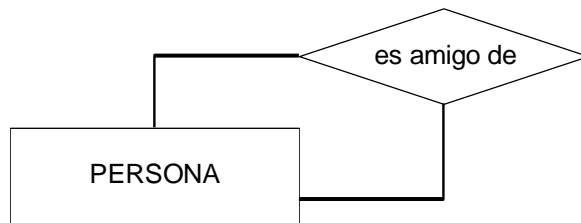
esto permite a cada ocurrencia de la relación formar parte de un par de empleados, uno en el rol de jefe y otro en el rol de subordinado.

En este tipo de relaciones, la misma ocurrencia de una entidad puede desempeñar el mismo rol, pero no en el mismo momento.

Relaciones simétricas:

Para la mayoría de las relaciones recursivas, la distinción entre los distintos roles que una entidad puede desempeñar en la relación debe ser clara a efectos de que la relación esté definida correctamente.

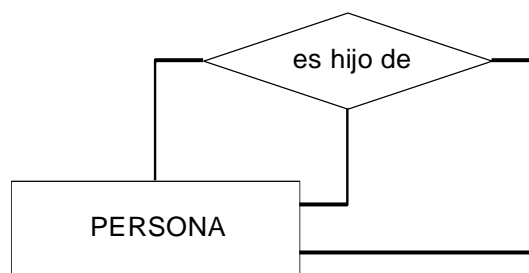
Para algunas relaciones recursivas, la posición de la entidad en la relación no es significativa. Estas relaciones se conocen como simétricas. La relación “es amigo de”, es un ejemplo de una relación simétrica.



Para este tipo de relaciones no es necesario distinguir el rol que juegan las entidades. Para el ejemplo anterior, podríamos indicar: *Persona* es amiga de *Persona*

Relaciones recursivas de orden superior

La relación de 3º orden que se indicó anteriormente (ver (*)), podría haberse modelado de la siguiente manera:



indicando: *Persona* es hija de *Persona*(madre) y *Persona*(padre)

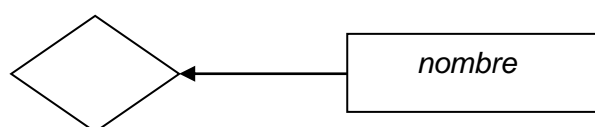
Conclusiones:

*Una relación es una posible asociación entre ocurrencias de diferentes entidades.
Puede considerarse como un patrón.
Muestran que un evento ocurre.*

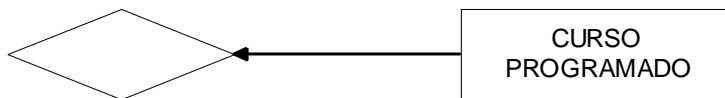
Entidad Asociativa

Una entidad asociativa es una entidad que actúa como relación y como entidad. Surge de una relación acerca de la cual se desea guardar información. Una ocurrencia de una entidad asociativa no existe si no existen las ocurrencias de las entidades que intervienen en la relación de la cual surge. La inversa no se cumple.

Se representa con un rectángulo, un nombre y una flecha hacia la relación:



Entidad asociativa: ejemplo



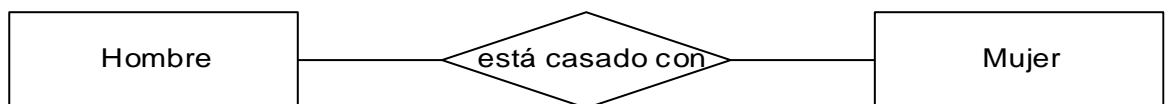
Una entidad asociativa actúa al mismo tiempo como una relación y como una entidad.

Como relación, indica que existe una asociación entre entidades, y tal como sucede con todas las relaciones, una ocurrencia de una entidad asociativa no puede existir sin que previamente existan las ocurrencias de las otras entidades.

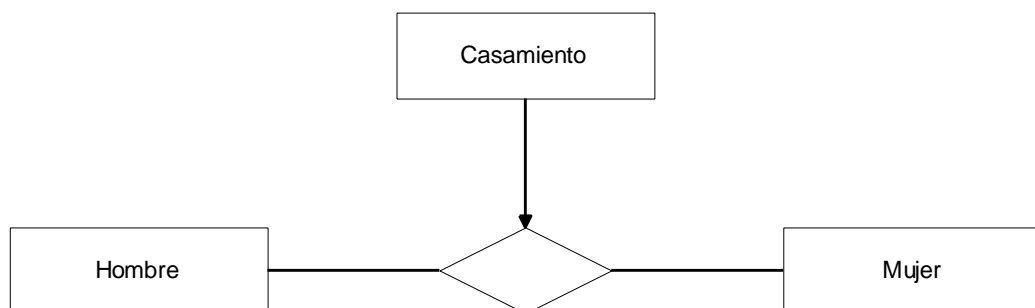
Una relación debería transformarse en una entidad asociativa si la relación posee atributos o bien la relación actúa como entidad en otras relaciones.

La entidad asociativa como una relación con atributos:

Los atributos de una entidad asociativa no describen a las entidades que participan de la relación, sino más bien a la asociación entre ellas. Por ejemplo, dada la siguiente relación:



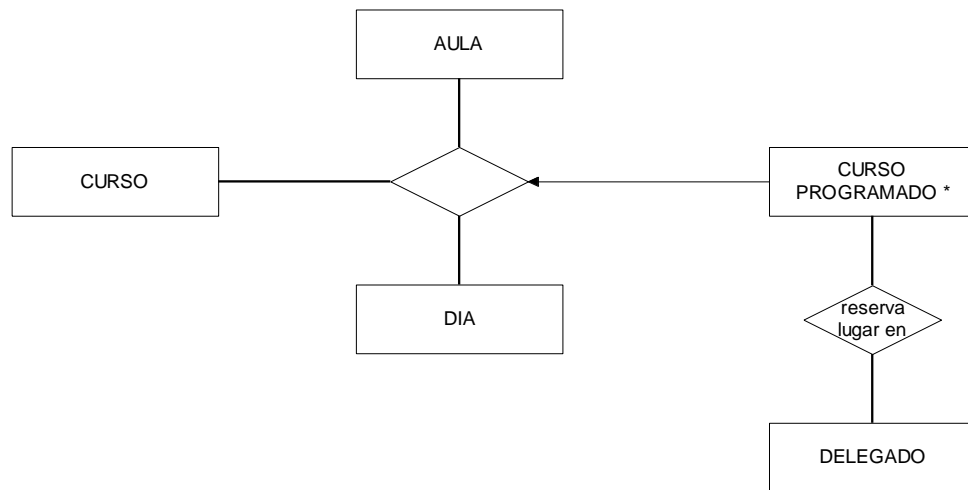
podríamos decir que es necesario recordar la fecha en la cual se casaron por Civil . Este atributo (fecha del Civil) no es un atributo ni de la entidad *Hombre*, ni de la entidad *Mujer*, sino que describe cuándo se estableció la ocurrencia de la relación. Para modelar esta situación, reemplazamos la relación “está casado con” por la entidad asociativa *Casamiento*, con “fecha del Civil” como atributo.



La entidad asociativa mantiene la propiedad de ser una relación (cada ocurrencia de *Casamiento* recuerda el hecho de que un determinado *Hombre* se casó con una determinada *Mujer* en una determinada fecha).

La entidad asociativa como entidad en otras relaciones:

Una entidad asociativa puede participar en relaciones con otras entidades. La entidad asociativa actúa como relación en el sentido de que “recuerda” la asociación original y también como una entidad involucrada como participante en relaciones posteriores. Por ejemplo, la entidad asociativa *Curso programado* actúa como una relación entre las entidades *Curso*, *Aula* y *Día*, y también como una entidad en la relación “reserva lugar en”:



Para que una ocurrencia de una entidad asociativa pueda participar en otra relación por sí misma, primero debe crearse. No tiene sentido pensar en un *Delegado* de una empresa reservando lugar en un *Curso programado* antes de que éste haya sido creado definiendo una asociación entre *Curso*, *Aula* y *Día*.

Muchas veces, una entidad asociativa es inicialmente identificada y modelada como una entidad. Una vez detectado su doble comportamiento, basta con transformarla a una entidad asociativa. Una buena pista para ver si una entidad es asociativa es verificar si la entidad sospechosa no puede existir independientemente de las otras entidades con las cuales participa en una relación.

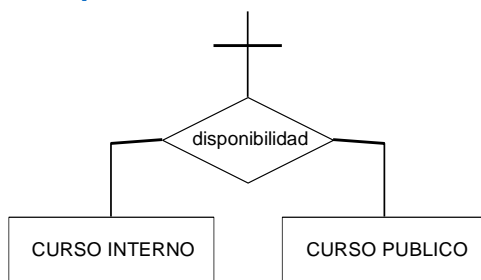
Conclusiones:

Una entidad asociativa es una entidad que actúa como relación y como entidad.

Surge de una relación acerca de la cual se desea guardar información.

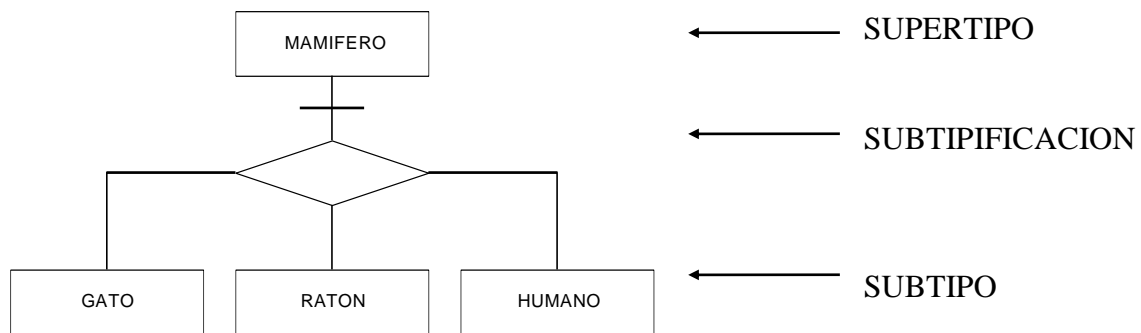
Una ocurrencia de una entidad asociativa no existe si no existen las ocurrencias de las entidades que intervienen en la relación de la cual surge. La inversa no se cumple.

Subtipo

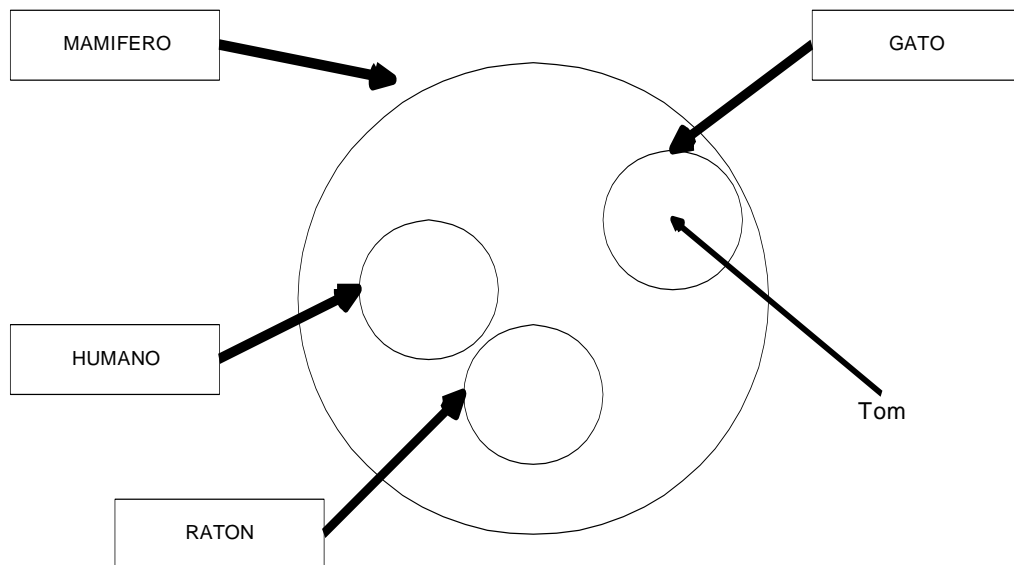


El subtipo de una entidad está formado por un grupo bien definido de ocurrencias de una entidad, que pueden considerarse como una entidad por derecho propio. Como ejemplo podemos considerar a la entidad *Mamífero*. Ocurrencias individuales de esta entidad podrían ser: Arquímedes, Pitágoras, Tom y Jerry.

Un grupo bien definido es la entidad *Humano*, que contiene varias ocurrencias (dos de las cuales se mencionaron en el párrafo anterior). Decimos que esta entidad es un subtipo de la entidad *Mamífero*. Otros subtipos de esta son *Gato* y *Ratón*.



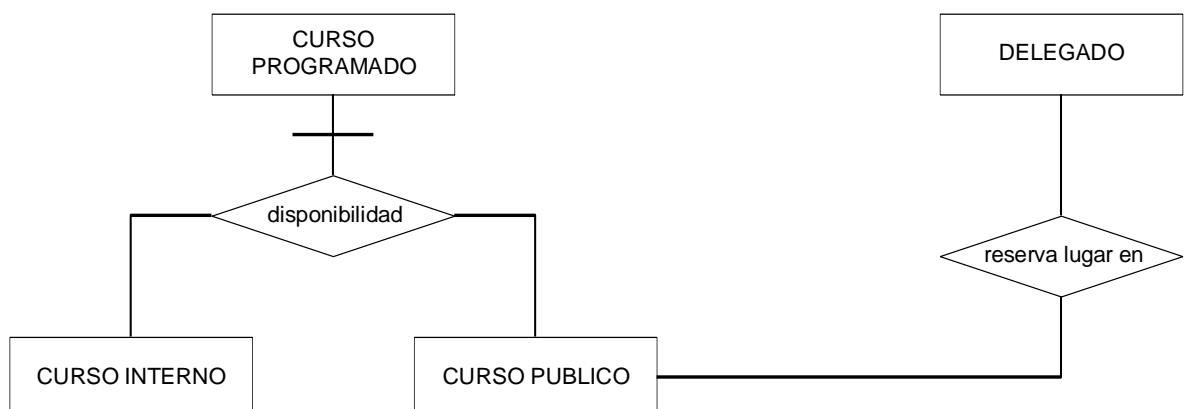
Esta subtipificación también podría visualizarse con un diagrama de Venn, que muestra a cada entidad como un subconjunto conteniendo ocurrencias de la misma:



Una ocurrencia del subtipo es siempre una ocurrencia del supertipo (Ej.: una ocurrencia de *Gato* es automáticamente una ocurrencia de *Mamífero*). Como consecuencia, cualquier propiedad que posea el supertipo es automáticamente una propiedad del subtipo. Podemos decir que el subtipo hereda las propiedades del supertipo.

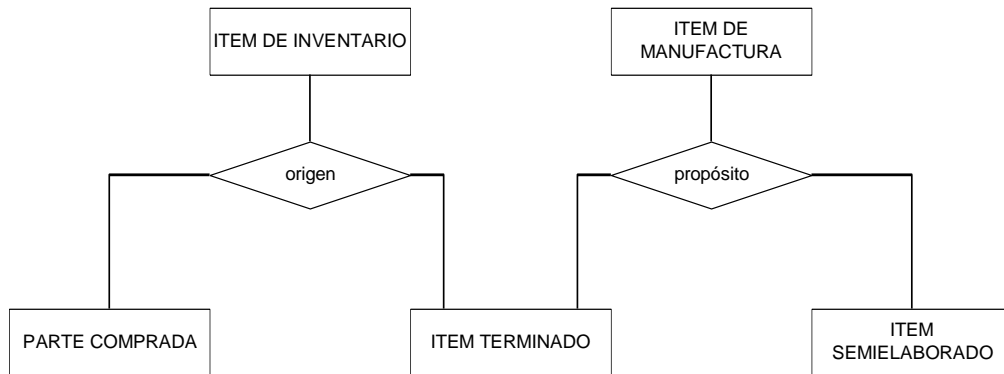
Además de las propiedades heredadas, un subtipo puede tener uno o ambos (pero nunca ninguno) de lo siguiente:

1. Atributos específicos: algún subtipo puede tener atributos que no son relevantes para otros subtipos. Esto se contrasta con los atributos comunes, que se declaran en el supertipo.
2. Relaciones específicas: algunas relaciones pueden afectar solamente a cierto subtipo de una entidad. Por ejemplo, un *Delegado* de una empresa sólo puede “reservar lugar” en un *Curso público*:



Múltiples supertipos:

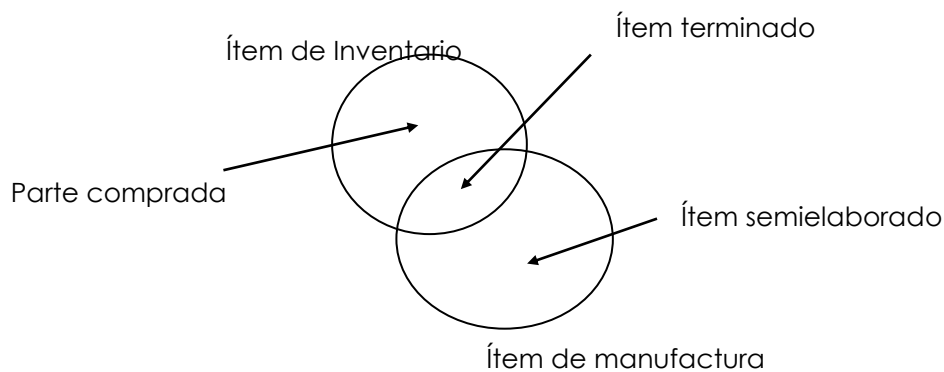
Una entidad puede ser un subtipo de subtipificaciones de varias entidades diferentes. En este caso, hereda cualquier relación y atributos comunes a ambos supertipos. Por ejemplo:



un *Ítem terminado* es un tipo de *Ítem de manufactura* y también un tipo de *Ítem de inventario*.

Como consecuencia, cualquier ocurrencia de *Ítem terminado* es también de *Ítem de manufactura* e *Ítem de inventario*, y por lo tanto, tiene atributos heredados de ambos.

Esto también puede mostrarse como un diagrama de Venn:

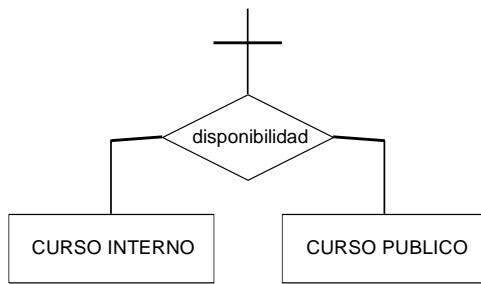


Conclusiones:

Un subtipo consiste en un subconjunto de ocurrencias de una entidad que puede considerarse como una entidad por derecho propio.

Lo importante: explicitar subtipos sólo si semánticamente es necesario.

Subtipificación



Una subtipificación indica que a la entidad involucrada se la considera como compuesta por grupos distintos y bien identificables, cada uno de los cuales se lo considera un subtipo.

Su importancia radica en la semántica, ya que permite que el modelador piense en conceptos generales, de alto nivel (relaciones, atributos) como relevantes para el supertipo y pueda detallar aquellos relacionados con cada uno de los subtipos, sin llegar a verse desbordado por la

complejidad.

Como un simple ejemplo, consideremos la subtipificación de la entidad *Mamífero* en *Gato*, *Ratón*, Esta subtipificación hace una distinción entre diferentes especies, a las que cada ocurrencia de un *Mamífero* pertenece, por lo tanto, podemos llamarla "especie". En este ejemplo, la subtipificación "especie" tiene 3 subtipos. En términos generales, una subtipificación puede tener varios subtipos.

La subtipificación genera subtipos mutuamente excluyentes:

Dentro de una subtipificación, no existe ocurrencia del supertipo que sea simultáneamente una ocurrencia de más de un subtipo de ella.

Alcance de una subtipificación:

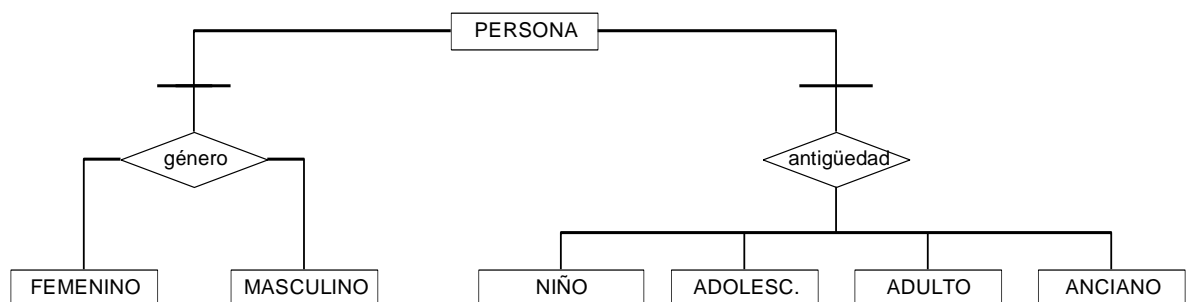
Decimos que una subtipificación está completa si cualquier ocurrencia del supertipo es una ocurrencia de exactamente uno de sus subtipos.

Algunas subtipificaciones poseen ocurrencias del supertipo que no caen en ninguna de las ocurrencias de los subtipos. Estos casos se los conoce como subtipificaciones parciales. Ejemplo de esto es la subtipificación "especie" que presentamos en párrafos anteriores.

Múltiples subtipificaciones:

Existen diferentes maneras para subtipificar una determinada entidad, de manera tal que cada subtipificación se identifica con un nombre único.

Por ejemplo: la entidad *Persona* podría subtipificarse por "género" en *Femenino* y *Masculino*; y por "antigüedad" en *Niño*, *Adolescente*, *Adulto* y *Anciano*:



Este DER debería leerse:

"Cada *Persona* puede ser *Femenino* o *Masculino*. La misma persona puede ser un *Niño*, o un *Adolescente*, o un *Adulto*, o un *Anciano*"

Ambas subtipificaciones del ejemplo son completas. Como consecuencia de ello podemos decir que:

"Cada *Persona* es, o bien *Femenino* o bien *Masculino*. La misma persona puede ser, o bien un *Niño*, o bien un *Adolescente*, o bien un *Adulto*, o bien un *Anciano*"

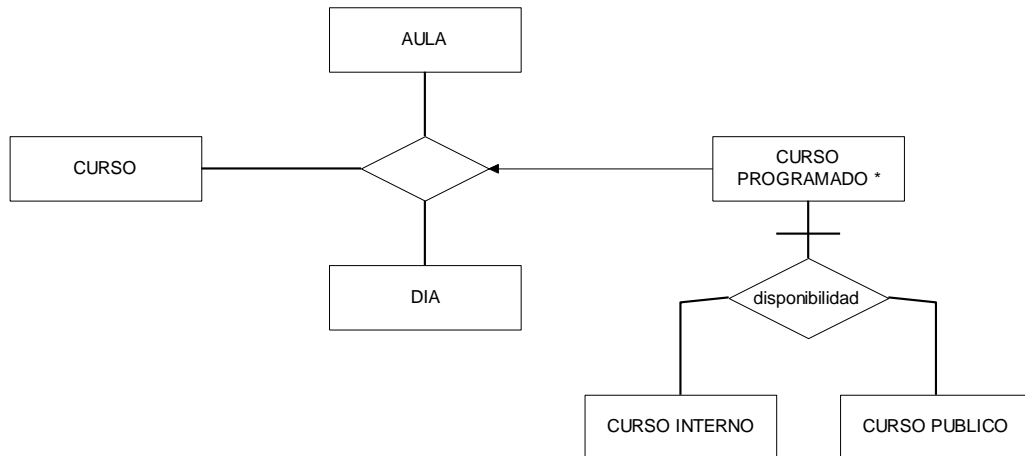
Subtipificaciones sin nombre:

El nombre de una subtipificación sirve para distinguirla, si es que existen más de una para una entidad. Una subtipificación de una entidad puede no tener nombre, sobre todo si el mismo nada aporta al porqué fue necesario distinguir a los subtipos.

Subtipificaciones de una entidad asociativa:

Si se define que una entidad asociativa va a tener subtipos, entonces, cada ocurrencia de uno de ellos debe ser una ocurrencia del supertipo. Cada ocurrencia del subtipo es, por lo tanto, una entidad asociativa, que hace referencia a las mismas entidades a las que hace referencia el supertipo.

Ejemplo:



Al momento de volcar cada una de estas entidades y relaciones al Diccionario de Datos tendríamos definiciones de:

- * *Curso programado* como entidad asociativa;
- * “disponibilidad” como una subtipificación;
- * *Curso público* como entidad asociativa, y
- “ *Curso interno* como entidad asociativa

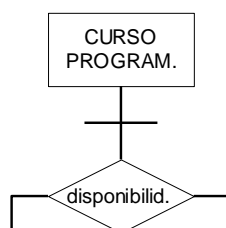
Conclusiones:

La subtipificación divide a una entidad en grupos bien definidos e identificables, cada uno de los cuales puede considerarse como una entidad.

Cada uno de esos grupos es mutuamente excluyente.

La subtipificación sólo debe aplicarse si semánticamente es necesario.

Supertipo



Podemos considerar a un supertipo como un agrupamiento general de varias entidades. Este agrupamiento es considerado como una entidad por derecho propio. Las entidades que son agrupadas dentro de esta entidad más general se conocen como subtipos.

Cuando una entidad posee subtipos, existen ciertas propiedades generales que son comunes con todos los subtipos:

1. Participar en relaciones: si una relación puede ocurrir para cualquiera de los subtipos de una entidad, entonces ocurre para el supertipo y se la dibuja en el DER vinculada con él.

2. Tener atributos comunes: los atributos comunes son aquellos atributos que siempre se encuentran en todos los subtipos de una entidad. Podemos decir que estos atributos “residen” en el supertipo cuando se lo especifica en el Diccionario de Datos.

Conclusiones:

Un supertipo es un agrupamiento general de entidades, al que puede considerarse como una entidad.

Contiene los atributos comunes a todos los subtipos que se desprenden de él.

Como detectar entidades y relaciones

Partiendo de la lista de requerimientos:

- Realizar una lista de sustantivos preliminar y luego ver si cada uno de ellos puede ser una entidad, es decir, si podemos caracterizarla con atributos, detectar al menos una ocurrencia, un identificador y sobre todo, si es significativa para el sistema.
- Detectar todas las posibles relaciones que vinculan entidades, enfocándose en los verbos que acompañan a los sustantivos.
- Realizar un DER a partir de los requerimientos, paso a paso, tratando de aplicar un análisis deductivo.

Veremos un ejemplo:

Supongamos que nuestro sistema en estudio se encarga de manejar la información de docentes, grupos, alumnos y trabajos prácticos de la materia Bases de Datos.

Listado Docentes = ${}_1\{\text{Apellido} + \text{Nombre} + \text{TE} + \text{Domicilio}\}_n$

Supongamos además que los Apellidos de los docentes no se repiten. Imaginemos además que esto es lo único que necesitamos del sistema.

El Listado de Docentes contiene entonces el apellido y nombre del docente, su dirección y su teléfono. Si tuviéramos que pensar que describen estos atributos contestaríamos al docente. Además el sistema necesita conservar estos datos para poder responder con el requerimiento.

Podríamos decir entonces que estamos en presencia de un tipo de objeto: DOCENTE.

Nuestro DER sería:

Docente

Docente = Apellido + Nombre + TE + Domicilio

Agreguemos un nuevo requerimiento en el que deseamos conocer cuáles son las comisiones en que se dicta nuestra materia y los docentes asignados a las mismas:

Listado De Comisiones = ${}_1\{\text{NroComisión} + \text{Aula} + {}_1\{\text{Apellido} + \text{Nombre}\}_n\}_n$

Nuestro DER tomaría la siguiente forma:



Estudiando este flujo de salida podríamos sugerir la idea de estar trabajando con docentes y con comisiones. Y también con la idea que el objetivo de este listado es mostrar quienes dictan clases en las comisiones.

Comisión = Comisión + Aula; el dato Comisión podría confundirse con el nombre del objeto, entonces lo cambiamos a NroComisión, lo mismo se debe hacer para con el flujo de datos de salida.

Supongamos que ahora tenemos un nuevo requerimiento, este nos dice que además de conocer quiénes son los docentes que dictan clase en las comisiones, también se desea saber que función cumplen, es decir, si ese docente en esa comisión es de práctica, de teoría o de practica/teoría.

El flujo de datos anterior toma la siguiente forma:

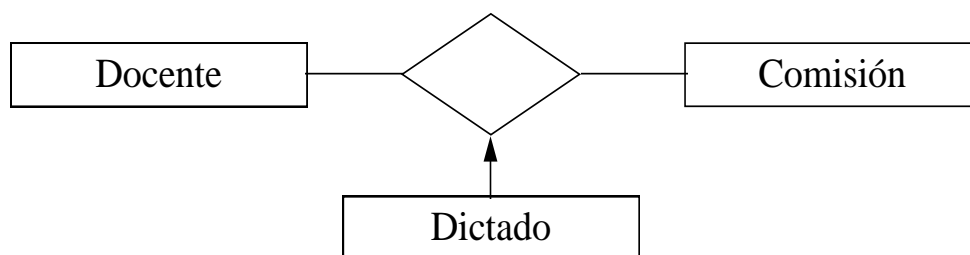
Listado De Comisiones = ${}_1\{\text{NroComisión} + \text{Aula} + {}_1\{\text{Apellido} + \text{Nombre} + \text{Función}\}_n\}_n$

Cuando evaluamos este atributo “Función” nos damos cuenta de que necesitamos más información.

Por ejemplo:

- Si cada docente solo desempeña una única función,
- o por el contrario, pueden desempeñar cualquier función.

Si contempláramos la primer alternativa veríamos que la función sería una característica exclusiva del docente, mientras que si observamos la segunda concluiríamos que esto de la función no es del docente, ni de la comisión solamente, sino que lo es de la relación entre el docente y la comisión. Este atributo debe ser agregado a la relación y por lo tanto toma la forma de Entidad o Tipo de objeto asociativo:



Algunas de las cosas que sufren cambios son: el nombre de la relación que se traslada de lugar y quizás toma un nombre más significativo, también aparece en el diccionario de datos.

Dictado = Apellido + NroComisión + Función

Supongamos que el entrevistado nos dice que: “...los números de comisión son únicos durante el año en vigencia; pero no así de un año al otro”.

Por ejemplo:

En el año 1996, la comisión 3-03 estuvo en el aula 210, la comisión 2-05, en el aula 205, la comisión 2-08, en el aula 208.

En el año 1997, la comisión 3-03 estuvo en el aula 211, la comisión 2-05 en el aula 205, la comisión 2-08 en el aula 208.

Es decir, vuelve a aparecer el mismo número de comisión en un aula diferente (o podría haber sido la misma). No se puede reconocer entonces cual es la comisión de un año en particular.

Nos está faltando un atributo: Año.

Quizás lo nombrado en los párrafos superiores quede expresado de la siguiente manera, alternativas:

- a) Una comisión queda identificada en forma única por el año y el número de comisión.
- b) Los números de las comisiones se repiten de un año a otro; pero no el mismo año o período.

Agregar este atributo basado en cualquiera de las visiones anteriores implicaría cambios en el diccionario de datos ya analizado y en el de las entidades o tipos de objetos.

Comisión = Año + Nro Comisión + Aula
Dictado = Año + Nro Comisión + Apellido + Función

El flujo de salida:

Listado De Comisiones=Año + ₁{NroComisión + Aula + ₁{Apellido + Nombre}_n}_n,

estamos por lo tanto manifestando que estamos obteniendo un listado de donde dictan clases los docentes de la cátedra durante un año determinado. Salvo que nuestra intención sea que el listado nos muestre donde dictan clases los docentes para todos los años; agregaríamos llaves alrededor de todo el flujo.

Con todos los cambios producidos por el atributo Año podemos concluir que los tipos de objetos asociativos arrastran en su definición de datos los atributos que identifican a cada una de las entidades que participan de la relación.

Evidentemente podríamos continuar suponiendo distintos casos agregando otros requerimientos y ajustarlos a visiones de contexto en algunos casos; pero en síntesis la cátedra pretende reconocer métodos alternativos de construcción y comprensión de los datos.

Cómo refinar el DER

1. Identificar relaciones de las cuales surjan entidades asociativas.
2. Identificar entidades que pueden llegar a ser asociativas.
3. Encontrar supertipos y subtipos si los hubiere.
4. Eliminar relaciones redundantes.
5. Identificar entidades poco significativas (que aparecen en una sola relación, unidas a entidades asociativas).
6. Detectar entidades asociativas seguidas y ver si se pueden unir.
7. Realizar el Diccionario de Datos de cada entidad.

Cómo realizar el Diccionario de Datos de Entidades que completa el DER

Se utiliza la misma sintaxis que para el diccionario de datos, el DER necesita que los atributos que caracterizan cada una de las entidades sean explicitados en esta herramienta.

El diccionario de datos suele emplear una notación conceptual adaptada de la forma Backus – Naur (BNF):

= significa *equivalente a*
+ significa *y*
{ }_n significa *iteraciones* de la componente encerrada, donde i indica el valor mínimo y n el valor máximo de la iteración
() significa *opcional*

Las entidades se especifican de la siguiente manera: se indica el nombre de la entidad, seguida de los atributos que la describen, empleando la notación formal del Diccionario de Datos.

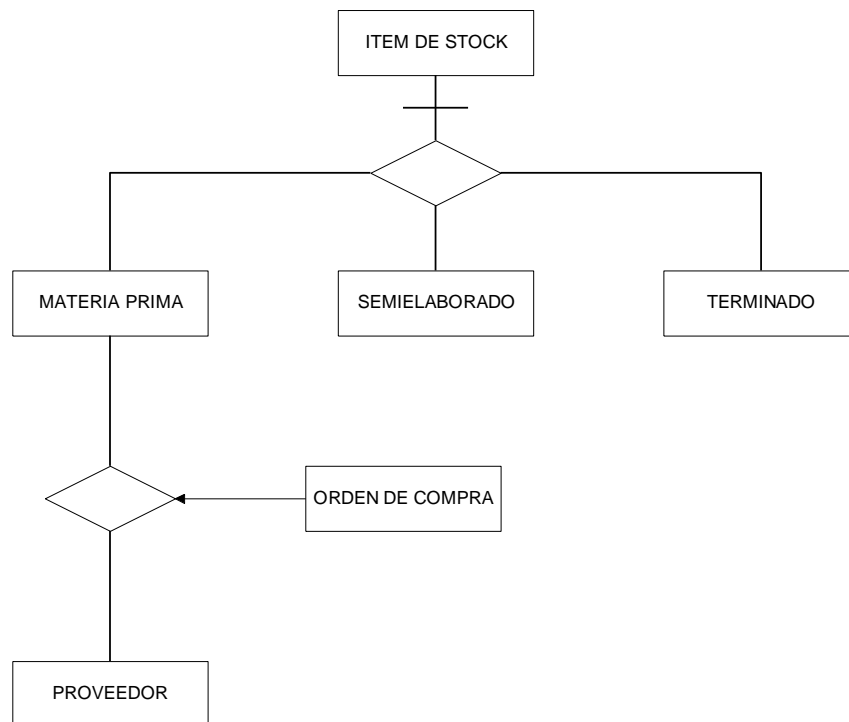
Para los supertipos sería conveniente listar sólo los valores comunes a los subtipos (ya que los valores propios de cada subtipo también son parte del supertipo).

Para los subtipos sería conveniente sólo los valores propios del subtipo (ya que los valores comunes que están en el supertipo y también son parte del subtipo).

Ejemplo:

Supongamos un sistema de Gestión de Stock de una empresa que manufactura los productos que vende.

DER (fragmento)



Diccionario de datos de Entidades

PROVEEDOR = código proveedor + domicilio + 0{teléfono}n

ORDEN DE COMPRA = número de orden + fecha + código proveedor + 1{código ítem + cantidad pedida}n

ÍTEM DE STOCK = código ítem + descripción + unidad de medida + stock

MATERIA PRIMA = presentación + lote de compra + stock mínimo + punto de pedido + precio de costo + ubicación en depósito + 1{código proveedor + precio proveedor}n

SEMIELABORADO = proceso actual

TERMINADO = precio de costo + margen de ganancia + precio de venta + presentación + ubicación en depósito

Cómo “mapear” el DER al gráfico que forma parte de un Mapa de Información Preliminar

Como dijimos anteriormente, el DER muestra los objetos que componen la memoria esencial y las relaciones existentes entre ellos. Vimos también que una relación establece una asociación entre una o varias ocurrencias de una entidad y una o varias ocurrencias de otra u otras entidades.

El **una o varias** se llama **cardinalidad**.

Vinculaciones

En el mundo real los datos están fuertemente relacionados entre sí. Por lo tanto, es válido el planteo de construir un Modelo de Datos atendiendo y explicitando esas vinculaciones. Existen dos tipos de vinculaciones:

- **Vinculaciones: INTRA-ENTIDADES:** es decir, las vinculaciones entre atributos de la misma entidad.
- **Vinculaciones INTER-ENTIDADES:** es decir, las vinculaciones entre miembros de entidades diferentes.

Vinculaciones INTRA-ENTIDADES

Para el estudio de las vinculaciones entre atributos de la misma entidad, consideraremos el concepto de *DEPENDENCIA FUNCIONAL* y el proceso de *NORMALIZACIÓN* que explicaremos más adelante.

Vinculaciones INTER-ENTIDADES

Para el caso de vinculaciones entre miembros de diferentes entidades, tendremos en cuenta la siguiente definición:

- **Vinculaciones 1 a 1**

PACIENTE <=====> HISTORIA CLINICA

Un paciente tiene una sola historia clínica, y una historia clínica pertenece a un solo paciente.

- **Vinculaciones 1 a N**

CLIENTE <=====>> CAJA DE AHORRO

Un cliente tiene varias caja de ahorros y una caja de ahorro pertenece a un único cliente.

- **Vinculaciones M a N**

ALUMNO <<=====>> MATERIA APROBADA

Un alumno tiene 0, 1 o varias materias aprobadas y una materias tiene 0, 1 o varios alumnos que la aprobaron.

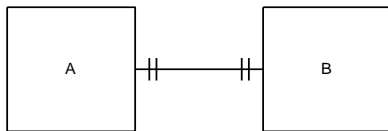
Un **Mapa de Información Preliminar** está compuesto por :

- un gráfico que muestra las entidades y las vinculaciones entre ellas, explicitando la cardinalidad.
- el Diccionario de Datos de Entidades, que se construyó con el DER.

En el gráfico, las entidades se muestran de la misma forma que el DER, es decir: con rectángulos identificados con el nombre de la entidad.

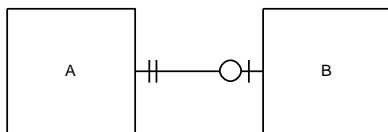
Las vinculaciones entre ellas pueden explicitarse de diferentes maneras, en este apunte se verá la notación empleada por Martin, a saber:

Vinculación 1 a 1 :



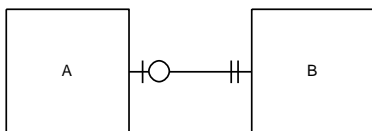
En todo instante, a cada ocurrencia de A le corresponde una y sólo una ocurrencia de B.
Y a cada ocurrencia de B le corresponde una y sólo una ocurrencia de A.

O bien :



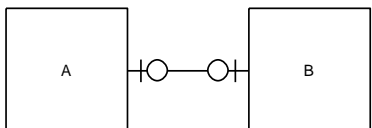
En todo instante, a cada ocurrencia de A le corresponde una o ninguna ocurrencia de B.
Y a cada ocurrencia de B le corresponde una y sólo una ocurrencia de A.

O bien :



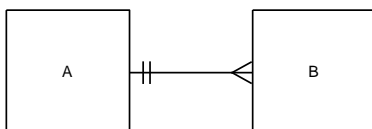
En todo instante, a cada ocurrencia de A le corresponde una y sólo una ocurrencia de B.
Y a cada ocurrencia de B le corresponde una o ninguna ocurrencia de A.

O bien :



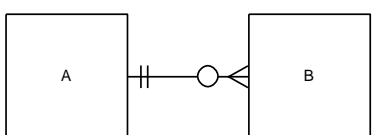
En todo instante, a cada ocurrencia de A le corresponde una o ninguna ocurrencia de B.
Y a cada ocurrencia de B le corresponde una o ninguna ocurrencia de A.

Vinculación 1 a N (muchos) :



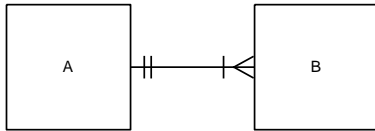
En todo instante, a cada ocurrencia de A le corresponden muchas ocurrencias de B.
Y a cada ocurrencia de B le corresponde una y sólo una ocurrencia de A.

O bien :



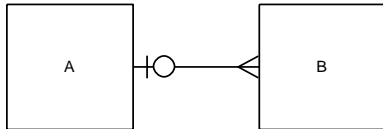
En todo instante, a cada ocurrencia de A le corresponde muchas o ninguna ocurrencia de B.
Y a cada ocurrencia de B le corresponde una y sólo una ocurrencia de A.

O bien :



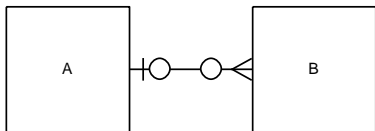
En todo instante, a cada ocurrencia de A le corresponde una o muchas ocurrencias de B.
Y a cada ocurrencia de B le corresponde una y sólo una ocurrencia de A.

O bien :



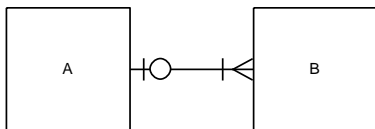
En todo instante, a cada ocurrencia de A le corresponden muchas ocurrencias de B.
Y a cada ocurrencia de B le corresponde una o ninguna ocurrencia de A.

O bien :



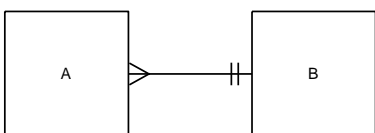
En todo instante, a cada ocurrencia de A le corresponde ninguna o muchas ocurrencias de B.
Y a cada ocurrencia de B le corresponde una o ninguna ocurrencia de A.

O bien :



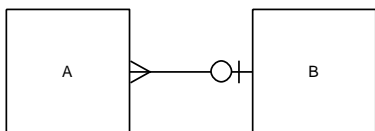
En todo instante, a cada ocurrencia de A le corresponde una o muchas ocurrencias de B.
Y a cada ocurrencia de B le corresponde una o ninguna ocurrencia de A.

Vinculación N (muchos) a 1:



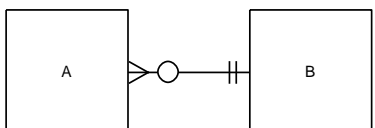
En todo instante, a cada ocurrencia de A le corresponde una y sólo una ocurrencia de B.
Y a cada ocurrencia de B le corresponden muchas ocurrencias de A.

O bien :



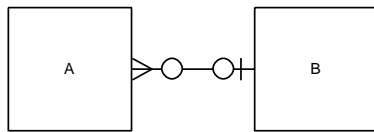
En todo instante, a cada ocurrencia de A le corresponde una o ninguna ocurrencia de B.
Y a cada ocurrencia de B le corresponden muchas ocurrencias de A.

O bien :



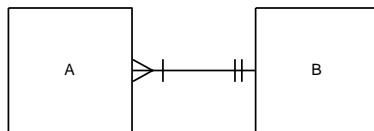
En todo instante, a cada ocurrencia de A le corresponde una y sólo una ocurrencia de B.
Y a cada ocurrencia de B le corresponden muchas o ninguna ocurrencia de A.

O bien :



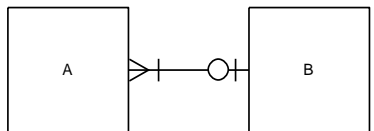
En todo instante, a cada ocurrencia de A le corresponden una o ninguna ocurrencia de B.
Y a cada ocurrencia de B le corresponden muchas o ninguna ocurrencia de A.

O bien :



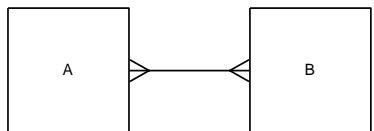
En todo instante, a cada ocurrencia de A le corresponde una y sólo una ocurrencia de B.
Y a cada ocurrencia de B le corresponde una o muchas ocurrencias de A.

O bien :



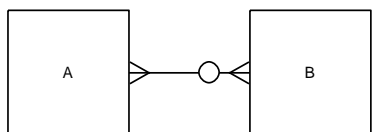
En todo instante, a cada ocurrencia de A le corresponde una o ninguna ocurrencia de A.
Y a cada ocurrencia de B le corresponde una o muchas ocurrencias de A.

Vinculación N a M (muchos a muchos) :



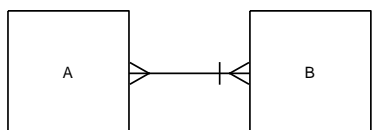
En todo instante, a cada ocurrencia de A le corresponden muchas ocurrencias de B.
Y a cada ocurrencia de B le corresponden muchas ocurrencias de A.

O bien :



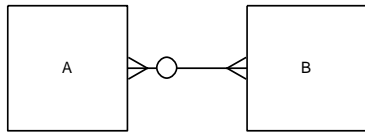
En todo instante, a cada ocurrencia de A le corresponde muchas o ninguna ocurrencia de B.
Y a cada ocurrencia de B le corresponden muchas ocurrencias de A.

O bien :



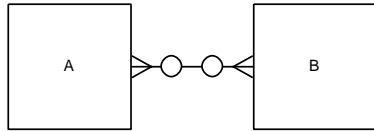
En todo instante, a cada ocurrencia de A le corresponde una o muchas ocurrencias de B.
Y a cada ocurrencia de B le corresponden muchas ocurrencias de A.

O bien :



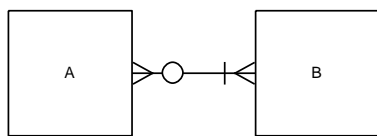
En todo instante, a cada ocurrencia de A le corresponden muchas ocurrencias de B.
Y a cada ocurrencia de B le corresponden muchas o ninguna ocurrencia de A.

O bien :



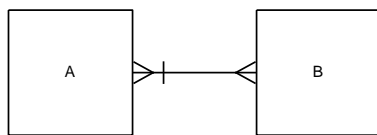
En todo instante, a cada ocurrencia de A le corresponden muchas o ninguna ocurrencias de B.
Y a cada ocurrencia de B le corresponden muchas o ninguna ocurrencia de A.

O bien :



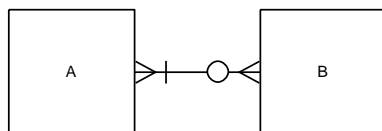
En todo instante, a cada ocurrencia de A le corresponde una o muchas ocurrencias de B.
Y a cada ocurrencia de B le corresponden muchas o ninguna ocurrencia de A.

O bien :



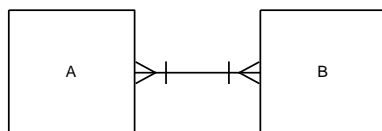
En todo instante, a cada ocurrencia de A le corresponden muchas o ninguna ocurrencias de B.
Y a cada ocurrencia de B le corresponde una o muchas ocurrencias de A.

O bien :



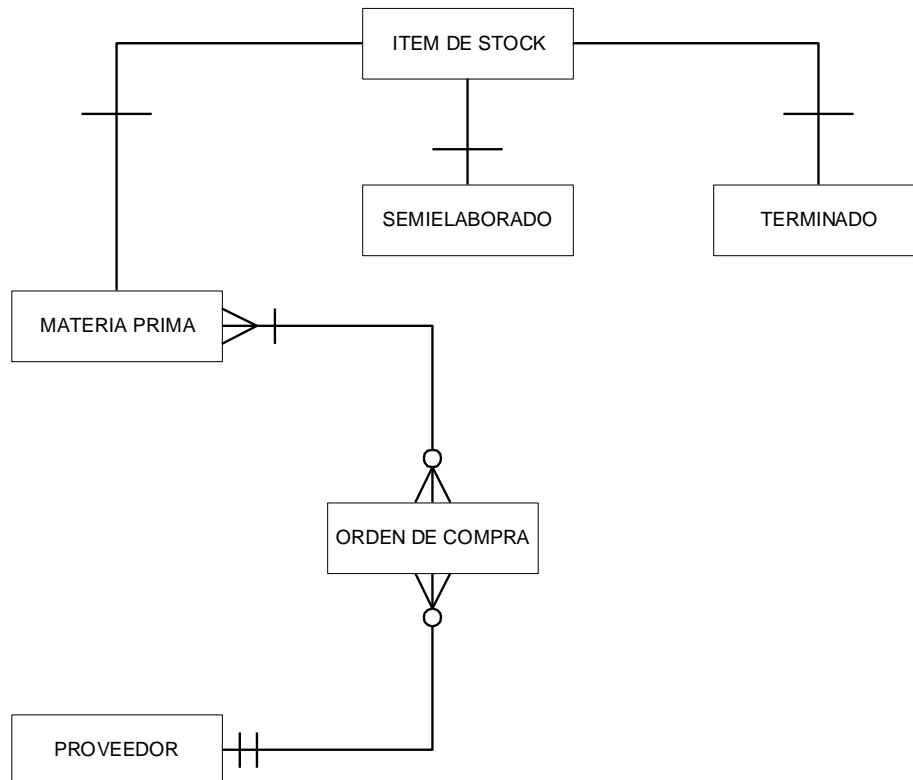
En todo instante, a cada ocurrencia de A le corresponden muchas o ninguna ocurrencias de B.
Y a cada ocurrencia de B le corresponde una o muchas ocurrencias de A.

O bien :



En todo instante, a cada ocurrencia de A le corresponde una o muchas ocurrencias de B.
Y a cada ocurrencia de B le corresponde una o muchas ocurrencias de A.

Veamos cómo queda el gráfico del Mapa de Información Preliminar de nuestro fragmento de DER del ejemplo del sistema de Gestión de Stock presentado cuando tratamos la construcción del Diccionario de Datos:

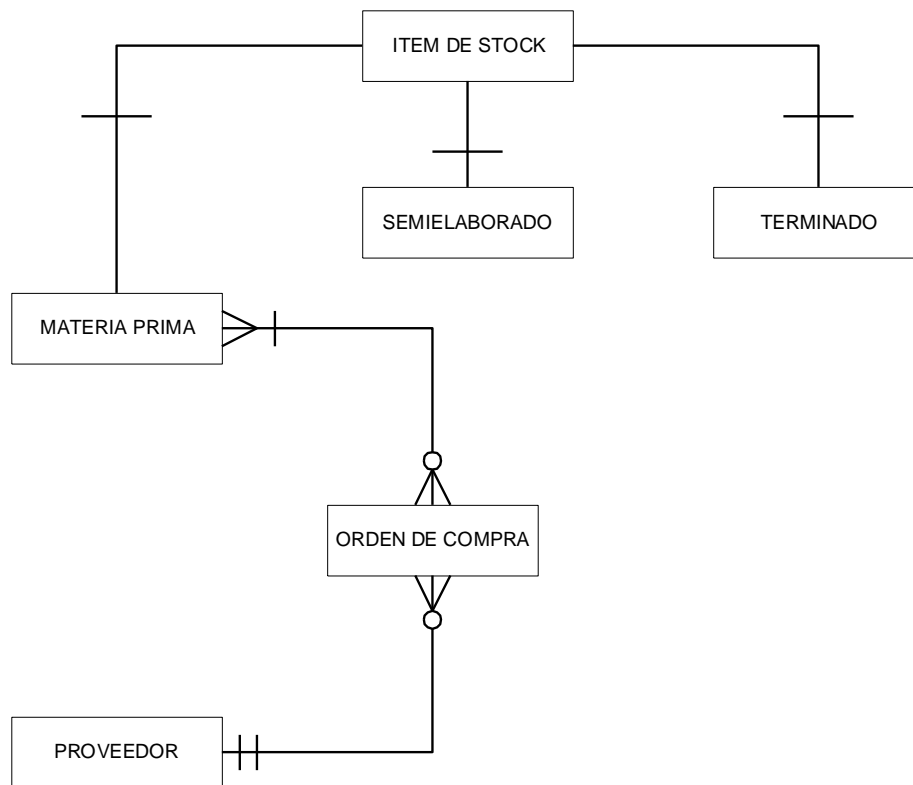


Veamos qué nos están diciendo las vinculaciones :

- Dada una orden de compra, ésta puede ser para una materia prima o para varias.
- Una materia prima puede haber sido pedida en más de una orden de compra o todavía en ninguna.
- Una orden de compra es para un único proveedor.
- Un proveedor puede tener varias órdenes de compra o bien ninguna.
- Materia prima, Semielaborado y Terminado son subtipos de Ítems de stock.

Podemos observar además, que las relaciones de subtipo se corresponden uno a uno con las vinculaciones de nuestro gráfico. Pero ¿por qué la relación de la cual surge la entidad Orden de Compra se mapea en **dos** vinculaciones ? Porque Orden de Compra es una entidad asociativa que involucra a **dos** entidades : Materia Prima y Proveedores. No olvidemos que una entidad asociativa muestra que un evento ocurre, relacionando a dos o más entidades y además el sistema deberá recordar datos de esa relación. La entidad asociativa contendrá datos comunes con todas las entidades que participan de la relación de la cual surgen.

El Mapa de Información preliminar de nuestro ejemplo será :



Diccionario de datos de Entidades

PROVEEDOR = código proveedor + domicilio + 0{teléfono}n

ORDEN DE COMPRA = número de orden + fecha + código proveedor + 1{código ítem + cantidad pedida}n

ÍTEM DE STOCK = código ítem + descripción + unidad de medida + stock

MATERIA PRIMA = presentación + lote de compra + stock mínimo + punto de pedido + precio de costo + ubicación en depósito + 1{código proveedor + precio proveedor}n

SEMIELABORADO = proceso actual

TERMINADO = precio de costo + margen de ganancia + precio de venta + presentación + ubicación en depósito

Optimización del Mapa preliminar

Como vimos anteriormente, con el Gráfico de vinculaciones más el Diccionario de Datos obtenemos un Mapa de Información Preliminar, donde explicitamos las vinculaciones entre las entidades.

El próximo paso es optimizarlo para que pueda cumplir los objetivos de todo Modelo de datos.

Antes de comenzar a describir cómo optimizar, será necesario volcar algunos conceptos o definiciones que intervienen en este proceso.

Algunos conceptos previos a la Normalización

Clave candidata

Es posible que existan más de un atributo o conjunto de atributos que puedan identificar unívocamente a una ocurrencia de una entidad. A estos se los llama claves candidatas. Dicho de otra manera son todas posibles claves de una entidad.

Ejemplo:

numero legajo al igual *tipo documento* + *numero documento* podrían ser pensadas, bajo ciertas condiciones como claves candidatas de la entidad *Alumno*.

Clave primaria

En el caso de que haya varias claves candidatas, se elige una como identificador principal y a esa se la llama clave primaria. Para indicarla en el diccionario de datos se la subraya.

Ejemplo:

Se puede elegir *numero legajo* como clave primaria de la entidad *Alumno*.

Clave alternativa

El resto de las claves candidatas que no hayan sido elegidas como clave primaria, se denominan claves alternativas.

Ejemplo:

tipo documento + *numero documento* podría constituir una clave alternativa de la entidad *Alumno*.

Dependencia Funcional

Para el estudio de las vinculaciones que existen entre los atributos que componen una entidad, aplicaremos el concepto de dependencia funcional.

Sea una entidad E y sus atributos X e Y ($E = X + Y$), decimos que X determina funcionalmente a Y sí y solo sí, para cada valor de X existe un único valor de Y en todo instante.

En otras palabras, dado el valor de X , queda determinado unívocamente el valor de Y .

Se indica:

$X \longrightarrow Y$

Ejemplo:

1. Sea la entidad *Empleado* = *numero legajo* + *nombre* y la siguiente restricción: *el sistema asignará un número de legajo único*.

Así, dado un número de legajo hay un único nombre de empleado que se corresponde con él.

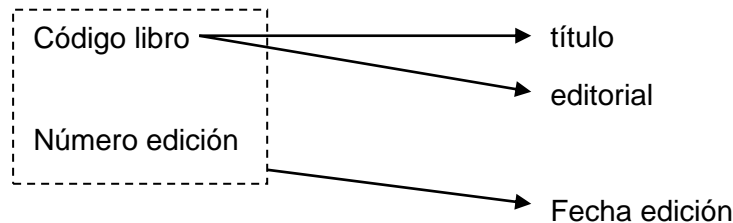
Esto se indica:

número legajo → nombre

2. Sea la entidad *Libro* = código libro + título + editorial + 1{número edición + fecha edición}_n y las siguientes restricciones:

- El sistema asignará un código de libro único.
- Un libro puede ser editado varias veces.
- Un libro es editado por una única editorial.
- Cada edición de un libro tiene una única fecha de edición.

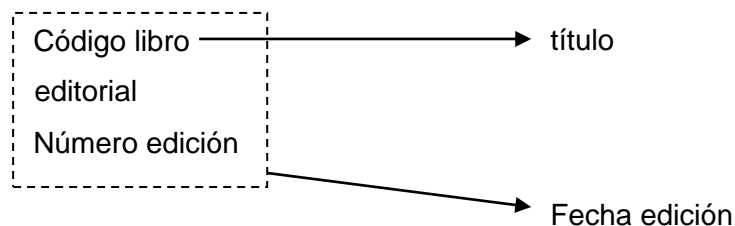
Por lo tanto:



Este esquema suele llamarse grafo de dependencias funcionales y representa las relaciones que existen entre los atributos de una entidad.

Supongamos ahora que modificamos la tercera restricción: *un libro puede ser editado por varias editoriales*. Tendremos los siguientes cambios:

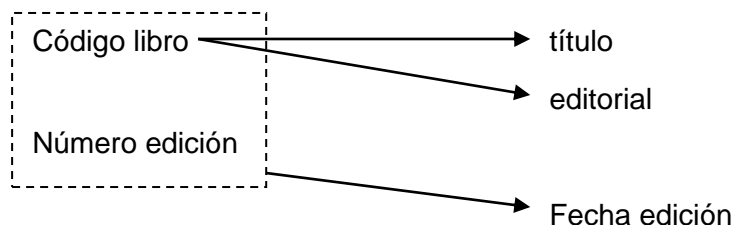
Libro = código libro + título + 1{editorial + 1{número edición + fecha edición}_n}



Importante: reconocer las dependencias funcionales entre atributos exige una previa explicitación del significado real de cada atributo.

Dependencias completas y no completas

Volvamos al ejemplo



- *Título* y *editorial* dependen funcionalmente de *código libro*,
- *Título* y *editorial* también dependen funcionalmente del par *código libro* y *número edición*, ya que para un valor de código libro + número edición, corresponde un único valor de título y editorial.

Por lo tanto:

- *Título y editorial* dependen funcionalmente **en forma completa** de *código libro*.
- *Título y editorial* dependen funcionalmente **en forma no completa** de *código libro* y *número edición*.

De ahora en adelante y salvo indicación expresa, cada vez que hablemos de dependencias funcionales, nos estaremos refiriendo a las que son completas.

Carácter unívoco

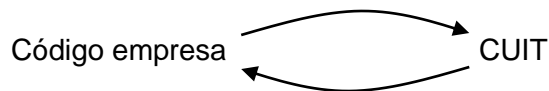
El concepto de dependencia funcional tiene carácter unívoco y no biunívoco.

Hay dependencia funcional entre *código libro* y *editorial*, porque a un valor de *código libro* le corresponde un único valor de *editorial*. Esto es suficiente para que exista dependencia funcional. Es evidente que aquí no aparece el carácter biunívoco, es decir, a una editorial le corresponden muchos código libro.

Supongamos la entidad:

Empresa = *código empresa* + *razón social* + *domicilio* + *CUIT* + *situación tributaria*

Como sabemos que el CUIT es único para cada empresa, así como el código empresa que a cada una le asigna el sistema, podemos graficar las dependencias funcionales entre código empresa y CUIT de la siguiente manera:

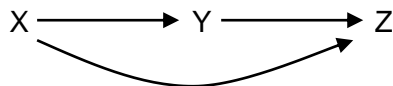


y aquí estamos frente a dos dependencias funcionales diferentes.

Transitividad

Sea la entidad E y sus atributos X, Y, y Z ($E = X + Y + Z$); donde X determina funcionalmente a Y, e Y determina funcionalmente a Z. Entonces, X determina funcionalmente en forma transitiva a Z.

En símbolos:



Ejemplo:

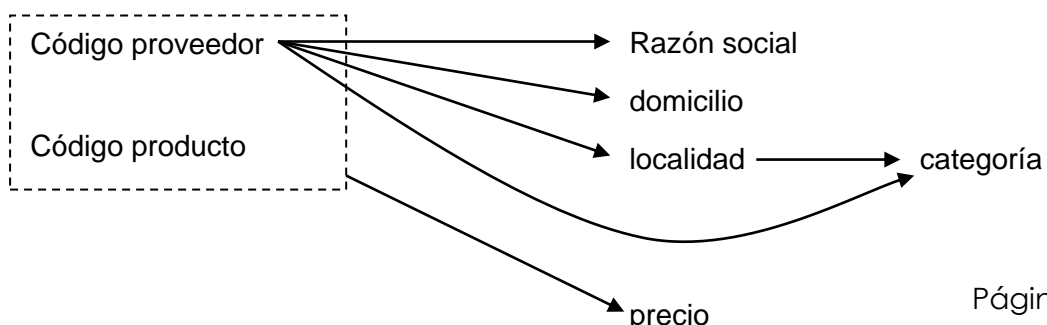
Supongamos que luego de construir el DER, nos queda una entidad:

Proveedor = *código proveedor* + *razón social* + *domicilio* + *localidad* + *categoría* + 1{*código producto* + *precio*}n

Y tenemos las siguientes restricciones:

- *El sistema asignará un código único al proveedor.*
- *Un proveedor vende varios productos y cada producto se identifica con un código.*
- *Un producto puede ser vendido por varios proveedores.*
- *Una localidad tiene una única categoría.*

El análisis de dependencias funcionales sería el siguiente:



Normalización

Hasta ahora nos hemos puesto de acuerdo en ciertas definiciones básicas y todavía no hemos abordado cómo optimizar nuestro modelo de datos, para que pueda satisfacer los objetivos:

1. Estructura mínima.
2. Que asegure la no pérdida de información.
3. Que facilite los procesos de mantenimiento (ABM).
4. Máxima supervivencia.

Es aquí donde aparece la normalización como herramienta para optimizar y poder lograr estos objetivos.

La normalización es una disciplina que, vía la aplicación sistemática de una sucesión de pasos formales, va transformando entidades originales en otras más adecuadas (en función de los objetivos), basándose en el significado de los datos o tratando de modelar su estructura real.

Esta disciplina es una buena ayuda para la optimización del modelo de datos pero no es la panacea.

Formas normales

Son los estados por los que va pasando una entidad durante esos sucesivos pasos de transformación. Así, se habla de una entidad en:

- 1ra forma normal (1FN)
- 2da forma normal (2FN)
- 3ra forma normal (3FN)
- Forma normal de Boyce-Codd (BCFN)
- 4ta forma normal (4FN)

Se dice que una entidad está en una determinada forma normal si las dependencias funcionales entre sus atributos cumplen ciertas condiciones. Cada forma normal representa un avance respecto de su forma normal predecesora.

Proceso de normalización

¿Qué haremos?

A partir de un caso particular:

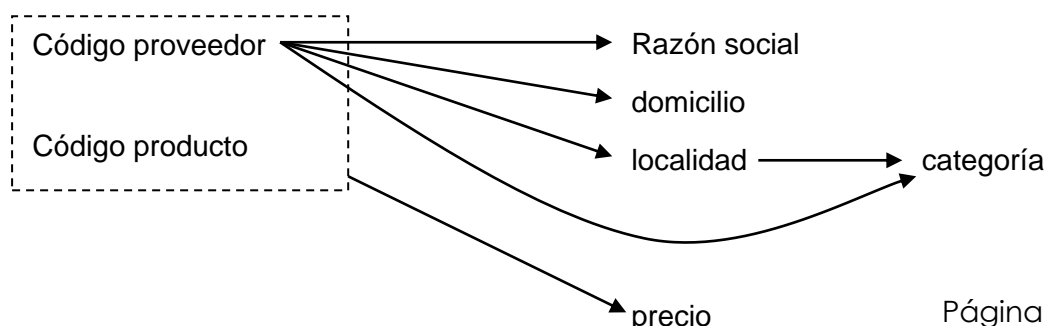
- Definiremos las tres primeras formas normales.
- Evaluaremos el cumplimiento de los objetivos.
- Definiremos un criterio para reconocer cuáles son las buenas descomposiciones.

Presentación del caso ejemplo

Sea la entidad:

Proveedor = código proveedor + razón social + domicilio + localidad + categoría + 1{código producto + precio}n

Y un conjunto de restricciones que determinan las siguientes dependencias funcionales:



Observaciones:

- Precio es el precio al que el proveedor vende cada producto.
- Categoría depende de la localidad donde se domicilia el proveedor.

Primera forma normal (1FN)

Una entidad está en primera forma normal si no tiene atributos multivalor.

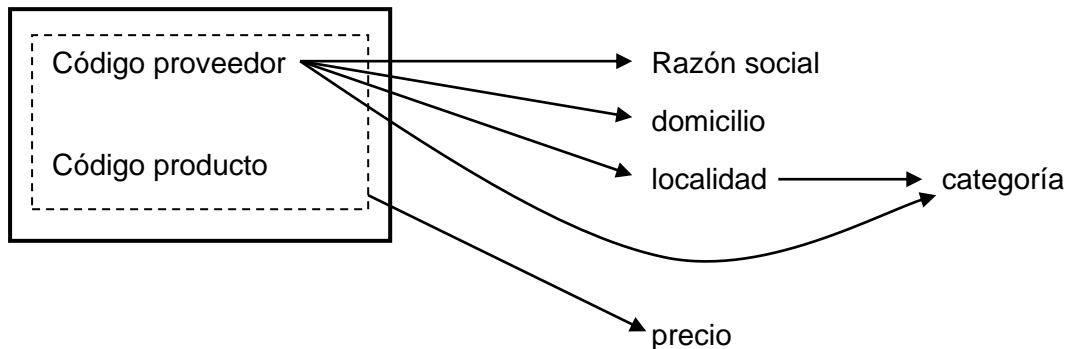
Esta definición está íntimamente ligada al concepto de **clave candidata**, ya que elegir una clave candidata asegura eliminar todos los atributos multivalor.

En nuestro caso, la única clave candidata es *código proveedor* + *código producto*. Eligiendo como clave candidata a este par de atributos, desaparecen los atributos multivalor. En cambio, si se hubiera elegido sólo *código proveedor* como clave, *código producto* + *precio* serían grupos repetitivos o atributos multivalor, es decir, para un *código proveedor* habría varios valores de *código producto*.

Por lo tanto, pasar de una entidad no normalizada a una entidad en 1FN implica elegir una clave que cumpla con la definición de clave candidata.

Observación:

Mostraremos resaltada la clave candidata en el grafo.

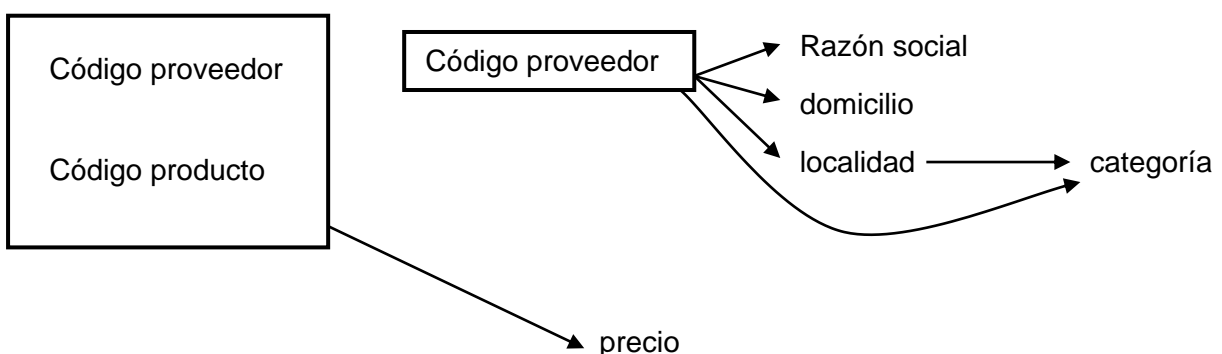


Segunda forma normal (2FN)

Una entidad está en segunda forma normal, sí y solo sí está en 1FN y todo atributo no clave depende de la clave en forma funcionalmente completa.

Es decir, se agrega la restricción de que toda dependencia funcional sea completa. Para lograr satisfacer esta restricción, es preciso descomponer la entidad en nuevas entidades.

En nuestro caso:



Al descomponer la entidad original en dos entidades, los atributos no clave de cada una de ellas, dependen de la clave y ahora en forma completa. Por lo tanto, están en 2FN.

Veámoslo a nivel del diccionario de datos:

Antes de normalizar

Proveedor = *código proveedor* + *razón social* + *domicilio* + *localidad* + *categoría* + 1{*código producto* + *precio*}*n*

1FN

Proveedor = *código proveedor* + *código producto* + *precio* + *razón social* + *domicilio* + *localidad* + *categoría*

2FN

Proveedor = *código proveedor* + *razón social* + *domicilio* + *localidad* + *categoría*

Lista de Precios Proveedor = *código proveedor* + *código producto* + *precio*

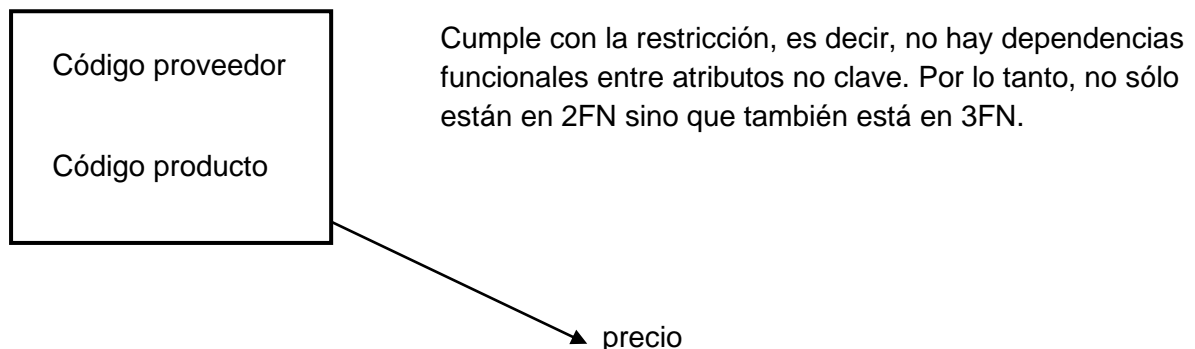
Podemos concluir que el proceso de normalización, al llegar a la 2FN termina “rompiendo” los grupos repetitivos, los cuales pasan a formar parte de una nueva entidad dentro de la cual ya no son más grupos repetitivos.

Tercera forma normal (3FN)

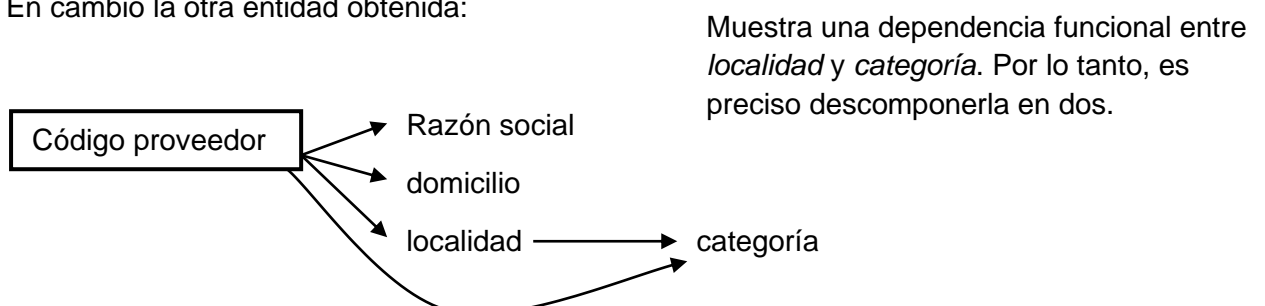
Una entidad está en tercera forma normal, sí y solo sí, está en 2FN y no existen dependencias funcionales entre atributos no clave.

Para lograr satisfacer esta restricción, es preciso descomponer nuevamente aquellas entidades que no la cumplan.

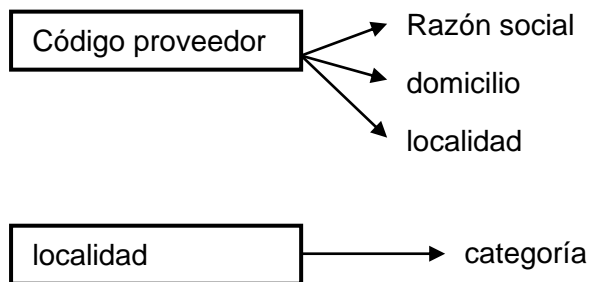
Veamos nuestro caso:



En cambio la otra entidad obtenida:

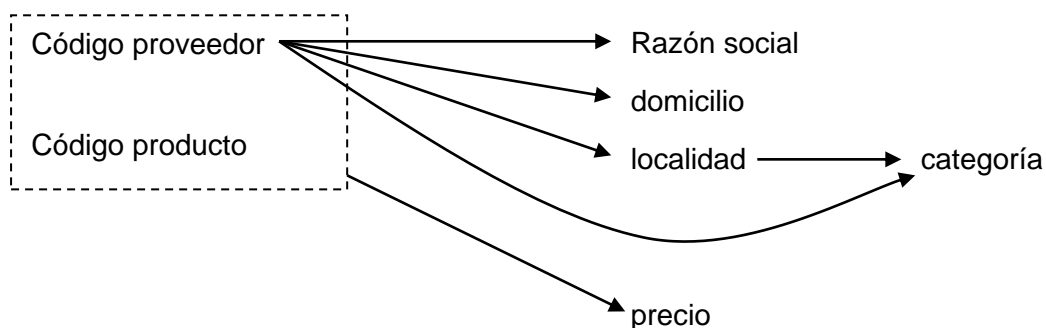


Este es el resultado de la descomposición:

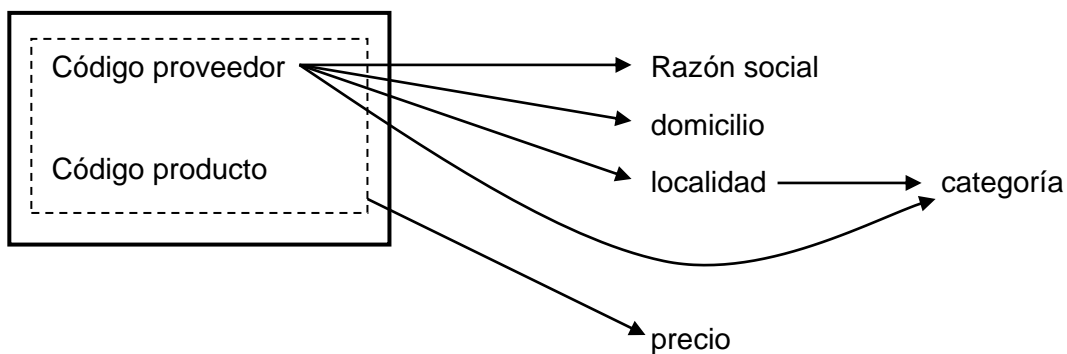


Cómo evolucionó la entidad de nuestro caso

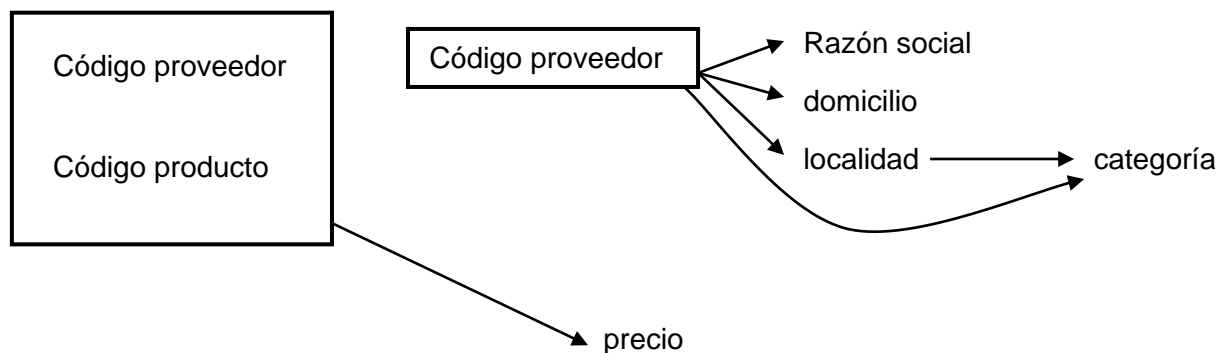
La entidad original de nuestro caso evolucionó de la siguiente manera:



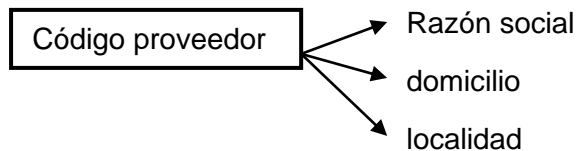
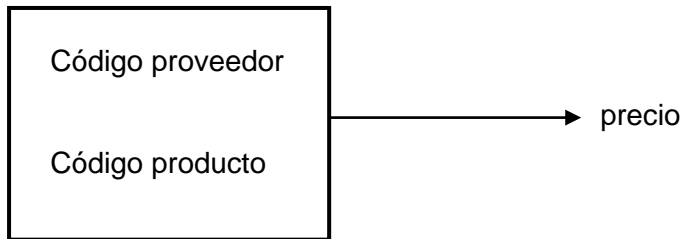
SE ELIGE UNA CLAVE (1FN)



SE BUSCAN DEPENDENCIAS COMPLETAS (se rompen los grupos “repetitivos”) (2FN)



SE ELIMINAN DEPENDENCIAS ENTRE ATRIBUTOS NO CLAVE (3FN)



La entidad original no normalizada, fue sustituida sin pérdida de información por tres entidades en 3FN, más adecuada desde el punto de vista de los objetivos.

Observación

De lo visto hasta aquí, podemos concluir que:

Una entidad está en 3FN, sí y solo sí, en todo momento, está formada por:

- **Un identificador único o clave.**
- **“n” atributos funcionalmente dependientes de la clave en forma completa y mutuamente independientes entre sí.**

Porque conviene normalizar

A medida que abordamos las tres primeras formas normales vimos como la entidad original desnormalizada fue reemplazada sin pérdida de información, por una estructura más adecuada teniendo presente los objetivos que debería cumplir todo Modelo de Datos.

Pero ¿por qué más adecuada?. Veamos el caso presentado.

Para ello, representaremos la entidad en 1FN como una tabla:

Código Proveedor	Código Producto	Razón Social	Precio	Domicilio	Localidad	Categoría
100	20	Juan Pérez	12	Mitre 788	Rosario	20
100	30	Juan Pérez	15	Mitre 788	Rosario	20
100	40	Juan Pérez	20	Mitre 788	Rosario	20
200	20	Pedro Martínez	10	Salta 180	San Lorenzo	15
200	35	Pedro Martínez	18	Salta 180	San Lorenzo	15
200	40	Pedro Martínez	28	Salta 180	San Lorenzo	15
300	10	José Pesci	100	Italia 455	Rosario	20

Y centraremos nuestro análisis en el cumplimiento del tercer objetivo:

- Facilitar los procesos de mantenimiento (ABM)

Problemas con la primera forma normal

Altas

Supongamos que necesita decir que tengo un proveedor nuevo, código 400, razón social José Solís, de Córdoba. Hasta tanto no le compre algún producto no puedo incorporar esta información, ya que código producto es parte de la clave.

Por lo tanto, poseo información en la realidad que no existe en el Modelo de Datos.

Bajas

Supongamos que el proveedor 300 no provee más el producto 10. Si elimino esa ocurrencia, pierdo información respecto a su razón social y domicilio.

Modificaciones

Supongamos que el proveedor 200 se muda. Es preciso modificar 3 ocurrencias.

Por lo tanto, con la entidad en 1FN no se cumple el objetivo de minimizar la estructura de datos ni de facilitar los procesos de mantenimiento.

Veamos que sucede en 2FN

Código Proveedor	Código Producto	Precio
100	20	12
100	30	15
100	40	20
200	20	10
200	35	18
200	40	28
300	10	100

Código Proveedor	Razón Social	Domicilio	Localidad	Categoría
100	Juan Pérez	Mitre 788	Rosario	20
200	Pedro Martínez	Salta 180	San Lorenzo	15
300	José Pesci	Italia 455	Rosario	20
400	José Solís	Maipú 344	Córdoba	25

Los tres problemas anteriores se resolvieron:

- Puedo incorporar los datos del proveedor 400.
- Si el proveedor 300 deja de proveer el producto 10, no pierdo sus datos.
- Si el proveedor 200 se muda, solo debo modificar una ocurrencia.

Problemas con la segunda forma normal

Altas

No puedo informar que la categoría de Tucumán es 35 hasta no tener algún proveedor de Tucumán.

Bajas

Si elimino el proveedor 200, pierdo la información de que la categoría de San Lorenzo es 15.

Modificaciones

Supongamos que Rosario cambia de categoría. Debo modificar 2 ocurrencias.

Esto obliga a continuar buscando una estructura que cumpla mejor con los objetivos.

Veamos que sucede en 3FN

Código Proveedor	Código Producto	Precio
100	20	12
100	30	15
100	40	20

Código Proveedor	Razón Social	Domicilio	Localidad
100	Juan Pérez	Mitre 788	Rosario
300	José Pesci	Italia 455	Rosario
400	José Solís	Maipú 344	Córdoba

Localidad	Categoría
Rosario	20
San Lorenzo	15
Córdoba	25
Tucumán	35

Los tres problemas anteriores están resueltos:

- Puedo decir que la categoría de Tucumán es 35, aún sin tener proveedores de esa localidad.
- Puedo eliminar al proveedor 200, junto con todos los productos que me vende, si perder información sobre la categoría de San Lorenzo.
- Si Rosario cambia de categoría, puedo informarlo modificando una sola ocurrencia.

Conclusiones

Por qué, entonces conviene normalizar?

Porque obtengo un Modelo de Datos que satisface dos de los principales objetivos:

- **Estructura mínima.**
- **Simplificación de los procesos de mantenimiento.**

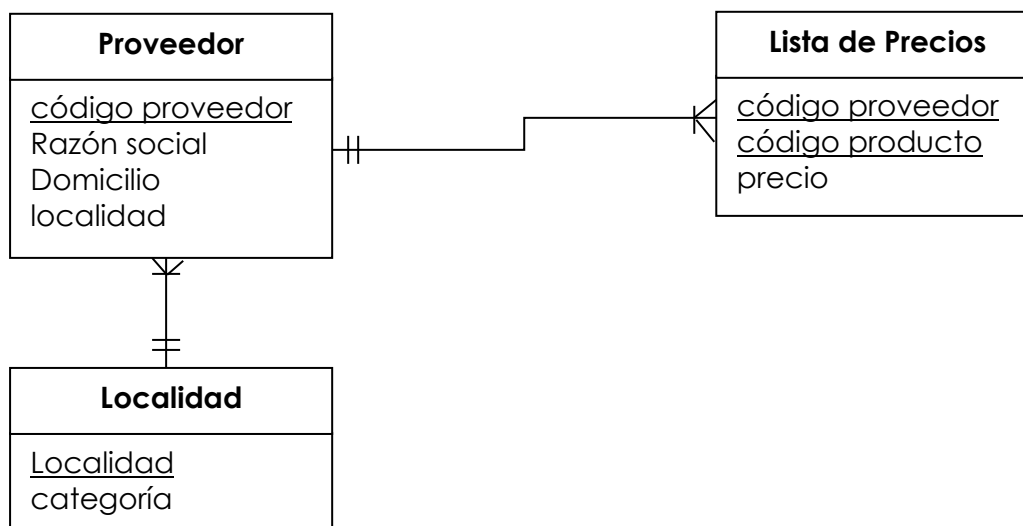
Respetando, además, el hecho de que **no hubo pérdida de información** de la estructura original en la estructura normalizada.

El cuarto objetivo: **máxima supervivencia**, quizás esté más asociado a una inteligente tarea de diseño en cuanto restricciones actuales y futuras. No obstante, es difícil tener en cuenta todas las restricciones asociadas a un problema real y tanto más difícil es considerar todas las que podrían aparecer con el transcurso del tiempo.

Mapa de Información Canónico (o Normalizado)

Es una herramienta gráfica que consiste en representar a cada entidad normalizada con un rectángulo, explicitando el nombre y los atributos que la componen, resaltando la clave, y cada vinculación entre las entidades.

Veamos cómo queda el mapa canónico de nuestro caso:



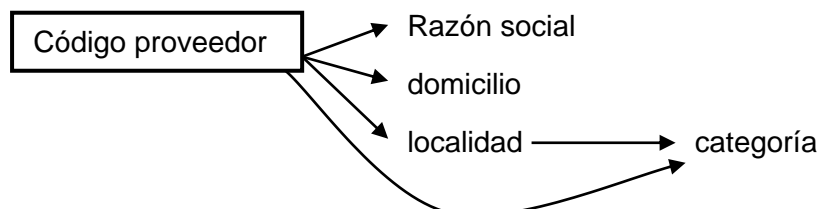
Las relaciones que generalmente aparecerán en nuestro mapa canónico, luego de haber normalizado el DER serán del tipo 1 a N. En caso de que llegaran a quedar entidades vinculadas 1 a 1, entonces debería unir las en una sola entidad.

Buenas Descomposiciones

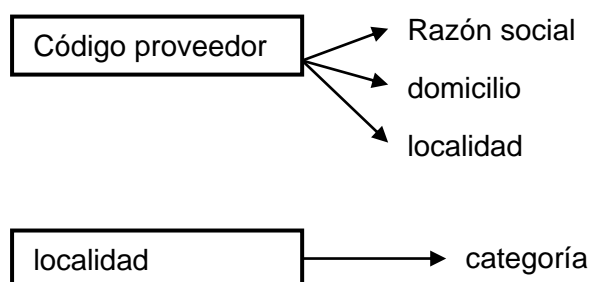
En el proceso de normalización, vimos que una vez obtenida la 1Fn (establecida la clave), ocurren una serie de descomposiciones.

El problema es cómo reconocer cuáles son buenas descomposiciones posibles.

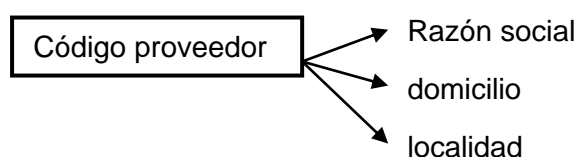
Ante la situación anteriormente vista:

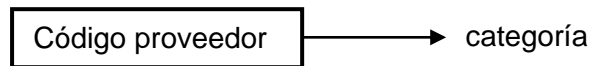


Al no estar en 3FN, la descompusimos de la siguiente manera (1):

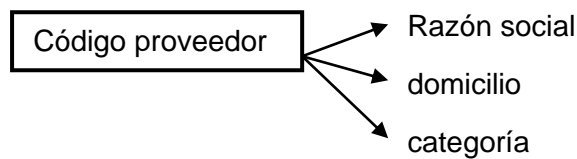


Pero, por qué de esa forma y no de esta (2):





O esta (3):



Para responder a estas preguntas veamos antes las siguientes definiciones:

En toda descomposición de una entidad en varias entidades, debe lograrse que las nuevas entidades sean independientes:

Un conjunto de entidades es independiente sí y solo sí cada una de ellas puede ser actualizada sin afectar necesariamente a las otras.

¿Cómo reconocer cuándo dos entidades son independientes?

Teorema

Dos entidades E1 y E2, producto de la descomposición de una entidad E son independientes, sí y solo sí, se cumplen las dos siguientes condiciones:

- Toda dependencia funcional en E puede ser lógicamente deducida de las dependencias funcionales en E1 y E2 más la vinculación entre E1 y E2.
- Los atributos comunes a E1 y E2 deben ser claves candidatas al menos de una de ellas.

Por lo tanto:

Descomposición	Condición A	Condición B	Conclusión
(1)	Se cumple.	Se cumple.	Buena descomposición.
(2)	No se cumple: la dependencia funcional entre localidad y categoría no se arrastra.	Se cumple.	Mala descomposición.
(3)	No se cumple: la dependencia funcional entre código proveedor y localidad no se arrastra.	No se cumple: el atributo común categoría no es clave de ninguna entidad	Mala descomposición.

Visiones para validar el Mapa de Información Canónico o Normalizado

Existen visiones que el modelo debe satisfacer. El análisis de dependencias funcionales implica:

- realizar una lectura inteligente del relevamiento identificando las visiones implícitas y explícitas.
- una explicación del significado real de cada atributo (exige explicitar la semántica del problema para que no haya ambigüedades).

Visiones de Contexto

En el relevamiento de los requerimientos del sistema surgen entre otras cosas, políticas, reglas, restricciones, condiciones, cotas y/o límites externos al sistema que pueden llegar a afectarlo, restringiendo, imponiendo o condicionando al mismo en forma directa o indirecta. A todas estas, se las conoce como **visiones de contexto**.

Las visiones de contexto son aquellas partes que describen como debe funcionar o realizarse una tarea o bajo qué condiciones, impidiendo o negando que pueda hacerse de otro modo. Son todas aquellas que hacen del sistema un sistema particular del negocio en el que estamos trabajando.

Si no hiciéramos el ejercicio de identificar e interpretar las visiones de contexto podríamos incurrir en el error de suponer que el sistema debe funcionar bajo cualquier variante, para todas las condiciones y en cualquier momento. Cuestión que no representa la realidad y por ello la modelamos.

Estas visiones nos definen como organizar la memoria esencial del sistema. Por ejemplo al decir que *un proveedor ofrece más de un producto*, estructuraremos los datos de forma diferente que si *un proveedor ofrece un único producto*.

Ejemplos de visiones de contexto:

- Una caja de ahorro pertenece a un solo cliente.
- Un crédito debe tener por lo menos un garante.
- Un ítem tiene un código único, es provisto por varios proveedores y cada uno a un precio diferente.
- La denominación de un ítem es única (no hay dos ítems con una misma denominación).
- Cada aeropuerto tiene un único código de aeropuerto No hay códigos repetidos y el nombre es único.
- El precio unitario de cada ítem de un proveedor varía de acuerdo a cierta cantidad mínima de compras establecidas por el propio proveedor.
- Cada proveedor nos hace una bonificación (un % de descuento) por cada ítem, independientemente de la cantidad que compremos.

Resumen

El modelado de datos es el proceso de representar y transformar un problema del mundo real en un sistema de información, usando bases de datos.

La primera fase del diseño de una aplicación (en el que una base de datos, generalmente, es parte), consiste en hablar con el cliente para saber qué quiere, y qué necesita realmente.

Una vez recogidos los datos, el siguiente paso es crear un modelo conceptual que represente el dominio del problema. El modelo más usado en bases de datos es el modelo Entidad-Relación, que es el que hemos explicado. Probablemente, “modelar” sea la parte más difícil de la resolución del problema, significa “pensar” en el problema y sus restricciones. Requiere el mayor esfuerzo intelectual del proceso, durante el cual, seguramente, deberemos tomar decisiones, que en cierto modo limitarán el modelo.

La siguiente fase es convertir el modelo conceptual en un modelo lógico. Existen varios modelos lógicos, pero el más usado, y el que por lo tanto explicamos aquí, es el modelo Relacional. La conversión entre el modelo conceptual y el lógico es algo bastante mecánico, aunque no por ello será siempre sencillo.

En el caso del modelo lógico relacional, existe un proceso que sirve para verificar que hemos aplicado bien el modelado, y en caso contrario, corregirlo para que sea así. Este proceso se denomina *normalización*.

Pasos propuestos para el modelado de datos:

1. Realizar el Relevamiento de la problemática.
2. Construir el modelo Entidad-Relación.
 - a. Identificar las entidades.
 - b. Identificar las asociaciones.
 - c. Identificar los atributos de las entidades y las asociaciones.
 - d. Eliminar conflictos semánticos.
 - e. Eliminar los atributos que pueden derivarse (es decir, que pueden obtenerse por cálculo).
3. Construir el Mapa de Información Preliminar.
4. Normalizar las entidades.
 - a. Eliminar entidades que estén íntegramente en otras y atributos innecesariamente repetidos. Construcción de nuevas entidades.
 - b. Verificar que existan los atributos para materializar las vinculaciones. Dos entidades vinculadas deben tener al menos un atributo en común que a su vez sea clave de una de las dos.
5. Construir el Mapa de Información Canónico o Normalizado.
 - a. Determinar vinculaciones inter entidades.
 - b. Validar las visiones de contexto.