

MODELOS DE DATOS Y MODELO ENTIDAD-RELACIÓN

1 DEFINICIÓN DE UN MODELO CONCEPTUAL	2
1.1 Modelo de Datos	2
1.2 Modelos Genéricos	3
2 EL MODELO RELACIONAL	4
2.1 Descripción	4
2.2 Entidad y Tipos de Entidades	4
2.3 Atributos	5
2.4 Relaciones. Cardinalidad. Tipos	5
2.4.1 Relación.....	5
2.4.2 Correspondencia y Cardinalidad.....	5
2.4.3 Tipos y Subtipos	6
a) Dependencia en existencia e identificación	7
b) Generalización-y Herencia.....	7
2.5 Claves. Tipos de Claves	9
2.6 Normalización. Formas Normales	9
2.6.1 Concepto de Normalización	9
2.6.2 Primera Forma Normal (1FN)	10
Definición	10
Algoritmo para pasar a 1FN:	10
Dependencia Funcional (DF).....	11
2.6.3 Segunda Forma Normal (2FN).....	11
Dependencia Funcional Completa (DFC)	11
Definición de 2FN.....	12
Algoritmo para pasar a 2FN:	12
2.6.4 Tercera Forma Normal (3FN).....	13
Dependencia Funcional Transitiva:.....	13
Algoritmo para pasar a 3FN	13
3 CONSTRUCCIÓN MODELO LÓGICO DE DATOS	14
3.1 Introducción	14
3.2 Transformación de diagramas E/R en relaciones.....	14
3.2.1 Transformación de Entidades	14
3.2.2 Transformación de Atributos de Entidades.....	14
3.2.3 Transformación de Relaciones	15
3.2.4 Transformación de atributos de relaciones.....	16

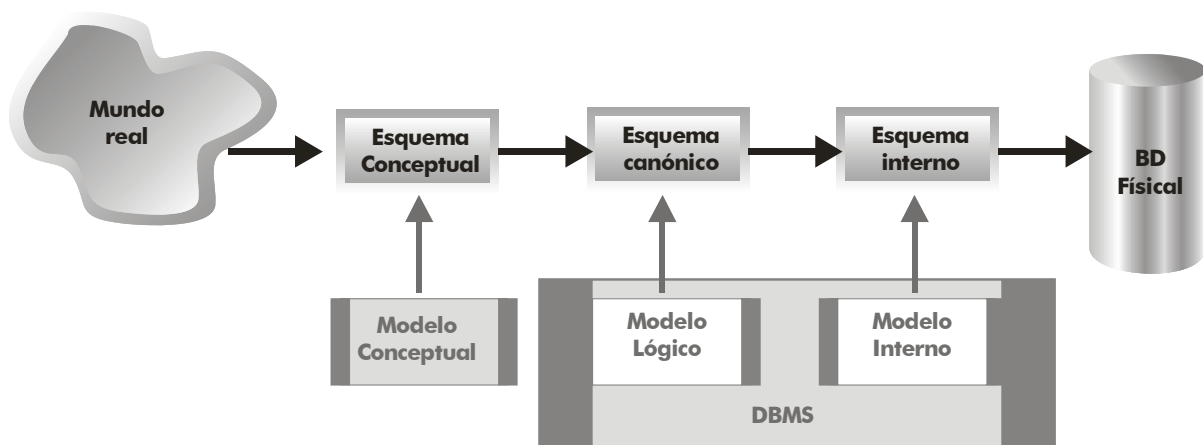
1 DEFINICIÓN DE UN MODELO CONCEPTUAL

1.1 Modelo de Datos

Los **modelos** se utilizan en todo tipo de ciencias. Su finalidad es la de simbolizar una parte del mundo real de forma que sea más fácilmente manipulable. En definitiva es un esquema mental (conceptual) en el que se intentan reproducir las características de una realidad específica.

En el caso de los **modelos de datos**, lo que intentan reproducir es una información real que deseamos almacenar en un sistema informático.

Se denomina esquema a una descripción específica en términos de un modelo de datos. El conjunto de datos representados por el esquema forma la base de datos.



En la ilustración anterior aparecen los distintos esquemas que llevan desde el mundo real a la base de datos física. Como se ve aparecen varios esquemas intermedios. Los que están más a la izquierda se alejan más de las características físicas. Los elementos de ese esquema son:

- **Mundo real.** Contiene la información tal cual la percibimos como seres humanos. Es el punto de partida
- **Esquema conceptual.** Representa el modelo de datos de forma independiente del DBMS que se utilizará.
- **Esquema canónico (o de base de datos).** Representa los datos en un formato más cercano al del ordenador
- **Esquema interno.** Representa los datos según el modelo concreto de un sistema gestor de bases de datos (por ejemplo Oracle)
- **Base de datos física.** Los datos tal cual son almacenados en disco.

Para conseguir estos esquemas se utilizan modelos de datos. El paso entre cada esquema se sigue con unas directrices concretas. Estas directrices permiten adaptar un esquema hacia otro.

1.2 Modelos Genéricos

Los dos **modelos fundamentales de datos son el conceptual y el lógico**. Ambos son conceptuales en el sentido de que convierten parámetros del mundo real en abstracciones que permiten entender los datos sin tener en cuenta la física de los mismos.

Diferencias entre los modelos de datos:

- El modelo conceptual es independiente del DBMS que se vaya a utilizar. El lógico depende de un tipo de SGBD en particular
- El modelo lógico es más cercano al ordenador
- Es más cercano al usuario el modelo conceptual, el lógico forma el paso entre el informático y el sistema.

Algunos ejemplos de modelos conceptuales son:

- Modelo Entidad/Relación (E/R)
- Modelo RM/T
- Modelos semántico

Ejemplos de modelos lógicos son:

- Modelo Relacional
- Modelo en Red
- Modelo Jerárquico

A continuación vamos a trabajar con el modelo conceptual más utilizado en el mundo de los SGBD: el modelo Entidad-Relación. Más adelante veremos la conversión de éste modelo conceptual, en el modelo lógico denominado Modelo Relacional.

2 EL MODELO RELACIONAL

2.1 Descripción

Fue ideado por **Peter Chen**ⁱ en los años 1976 y 1977 a través de dos artículos. Se trata de un modelo que sirve para crear esquemas conceptuales de bases de datos. De hecho es prácticamente un estándar para crear esta tarea.

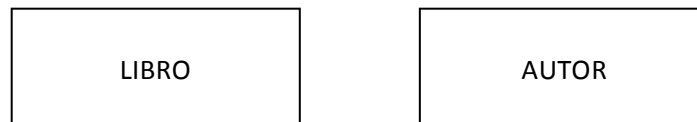
Se le llama **modelo E/R**ⁱⁱ e incluso **EI (Entidad / Interrelación)**. Sus siglas más populares son las **E/R** por que sirven para el inglés y el español.

Inicialmente (en la propuesta de Chen) sólo se incluían los conceptos de entidad, relación y atributos. Después se añadieron otras propuestas (atributos compuestos, generalizaciones,...) que forman el llamado **modelo entidad relación extendido** (se conoce con las siglas **ERE**)

2.2 Entidad y Tipos de Entidades

Se puede definir entidad como aquel objeto (real o abstracto) acerca del cual queremos almacenar información en la base de datos. Denominaremos a la estructura genérica en su sentido abstracto tipo de entidad, mientras que entidad será cada una de las ocurrencias o instancias de este tipo de entidad.

La **representación gráfica** de un tipo de entidad es un **rectángulo** etiquetado con el nombre del tipo de entidad:



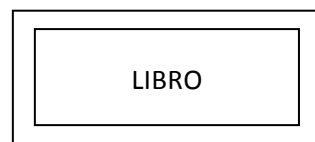
Tres reglas generales que debe cumplir cualquier entidad son (Tardieu):

- Tiene que tener existencia propia
- Cada ocurrencia de un tipo debe poder distinguirse de las demás
- Todas las ocurrencias de un tipo de entidad deben tener los mismos tipos de características (atributos).

Tipos de Entidades:

Existen dos clases de entidades: **regulares**, que tienen existencia por ellas mismas, y **débiles**, cuya existencia depende de otro tipo de entidad (p. ej., FAMILIAR depende de que exista EMPLEADO, y la eliminación de EMPLEADO obliga a la eliminación de FAMILIAR).

Los tipos de entidad débil se representan con dos rectángulos concéntricos con su nombre en el interior:



2.3 Atributos

Cada una de las propiedades o características que tiene un tipo de entidad o un tipo de relación se denomina ATRIBUTO. La representación gráfica de un atributo consiste en una elipse con el nombre del atributo en su interior o, mejor aún, un círculo junto a una etiqueta.



Entre todos los atributos de un tipo de entidad debemos elegir uno o varios que actúen como claves primarias. Estos atributos se representarán de la misma forma, pero se rellenará el círculo.

2.4 Relaciones. Cardinalidad. Tipos

2.4.1 Relación

Se entiende por relación a aquella **asociación o correspondencia existente entre entidades**. Llamaremos tipo de relación a la estructura genérica del conjunto de relaciones existentes entre dos o más tipos de entidad.

El tipo de relación se representa mediante un **rombo** etiquetado con el nombre de la relación, unido mediante arcos a los tipos de entidad que asocia. P. ej.:



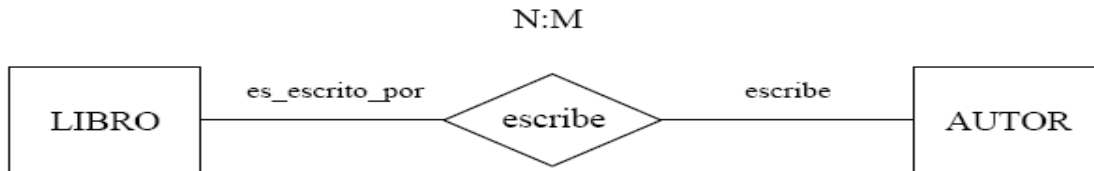
Entre dos tipos de entidades puede existir más de un tipo de relación.

Una relación se define por su nombre y por su grado. El nombre es el identificador que se le da a la propia relación, y el grado equivale al número de tipos de entidad a los que asocia o relaciona. Así por ejemplo, en el caso anterior, ‘escribe’ es una relación de grado 2. También pueden existir relaciones de grado 1 (o reflexivas), de grado 3, 4, ...

En general, las relaciones de grado superior a 2 se suelen reducir a relaciones de grado 2, si la semántica de la relación lo permite, puesto que se facilita tanto la comprensión del diagrama como la posterior conversión a otros modelos.

2.4.2 Correspondencia y Cardinalidad

Otro elemento que caracteriza a las relaciones es el tipo de correspondencia, que es el número máximo de ocurrencias de cada tipo de entidad que pueden intervenir en una ocurrencia del tipo de relación que se está tratando. Gráficamente, esto se representa con alguna de estas etiquetas textuales: **1:1**, **1:N**, **N:M**. El papel o rol es la función que cada tipo de entidad realiza en el tipo de relación. Se representa con su nombre sobre el arco que une el tipo de entidad con el tipo de relación. P. ej.:

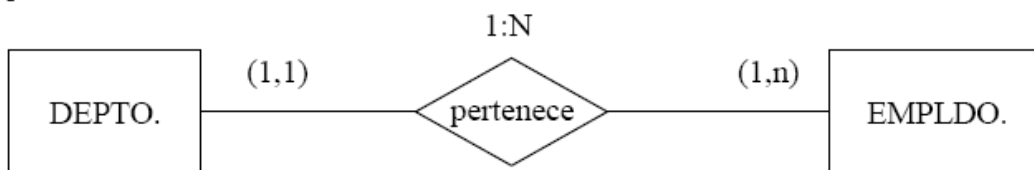


Cardinalidad de un tipo de entidad

Se define como el número máximo y mínimo de ocurrencias de un tipo de entidad que pueden estar relacionadas con una ocurrencia del otro u otros tipos de entidad que participan en la relación. Su representación gráfica es una etiqueta del tipo (0,1), (1,1), (0,n), o (1,n), según corresponda. El significado de cada una es:

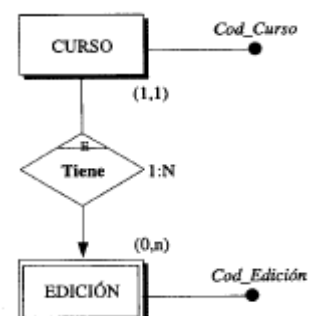
- **(0,1)**. Para una ocurrencia determinada de los otros tipos de entidades que participan en la relación, el valor mínimo de ocurrencias de la entidad que se trata es 0, y el número máximo es 1.
- **(1,1)**. Para una ocurrencia determinada de los otros tipos de entidades que participan en la relación, debe existir una y sólo una ocurrencia de la entidad que se trata.
- **(0,n)**. Para una ocurrencia determinada de los otros tipos de entidades que participan en la relación, el valor mínimo de ocurrencias de la entidad que se trata es 0, y el número máximo es n.
- **(1,n)**. Para una ocurrencia determinada de los otros tipos de entidades que participan en la relación, debe existir como mínimo una ocurrencia de la entidad que se trata, si bien no hay límite en el número de veces que dicha ocurrencia puede aparecer en la relación.

Ejemplo:



2.4.3 Tipos y Subtipos

En este apartado se verán las diferentes tipos de relaciones que propuso Chen (tipos) y las mejoras que sobre su modelo hicieron otros autores, en lo que se ha denominado

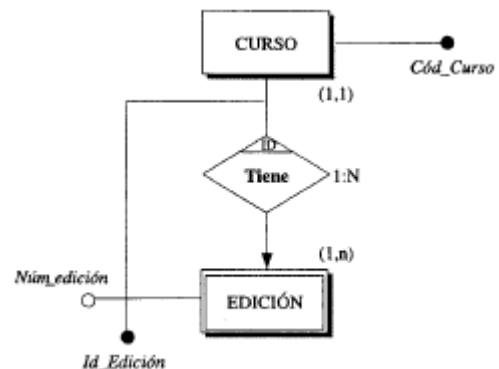


Modelo Entidad-Relación Extendido M-EE

a) Dependencia en existencia e identificación

Una entidad débil queda definida siempre a través de una interrelación especial que induce la dependencia de esta entidad de otra de orden superior (que puede ser entidad fuerte o débil). Toda entidad débil tiene una **dependencia en existencia** de la entidad de orden superior, definiéndose entre ellas una jerarquía de dos niveles.

Dependencia en identificación: Hay una dependencia en identificación cuando, además de la dependencia en existencia, las ocurrencias del tipo de entidad débil no se pueden identificar solo mediante sus propios atributos, sino que se tiene que añadir la clave de la ocurrencia de la entidad regular de la cual depende.



b) Generalización-y Herencia

La descomposición de tipos de entidad en varios subtipos es una necesidad muy habitual en el modelado de bases de datos. En el mundo real se pueden identificar varias jerarquías de entidades. La relación que se establece entre un supertipo y sus subtipos corresponde a la noción de ‘es_un’ o ‘es_un_tipo_de’.

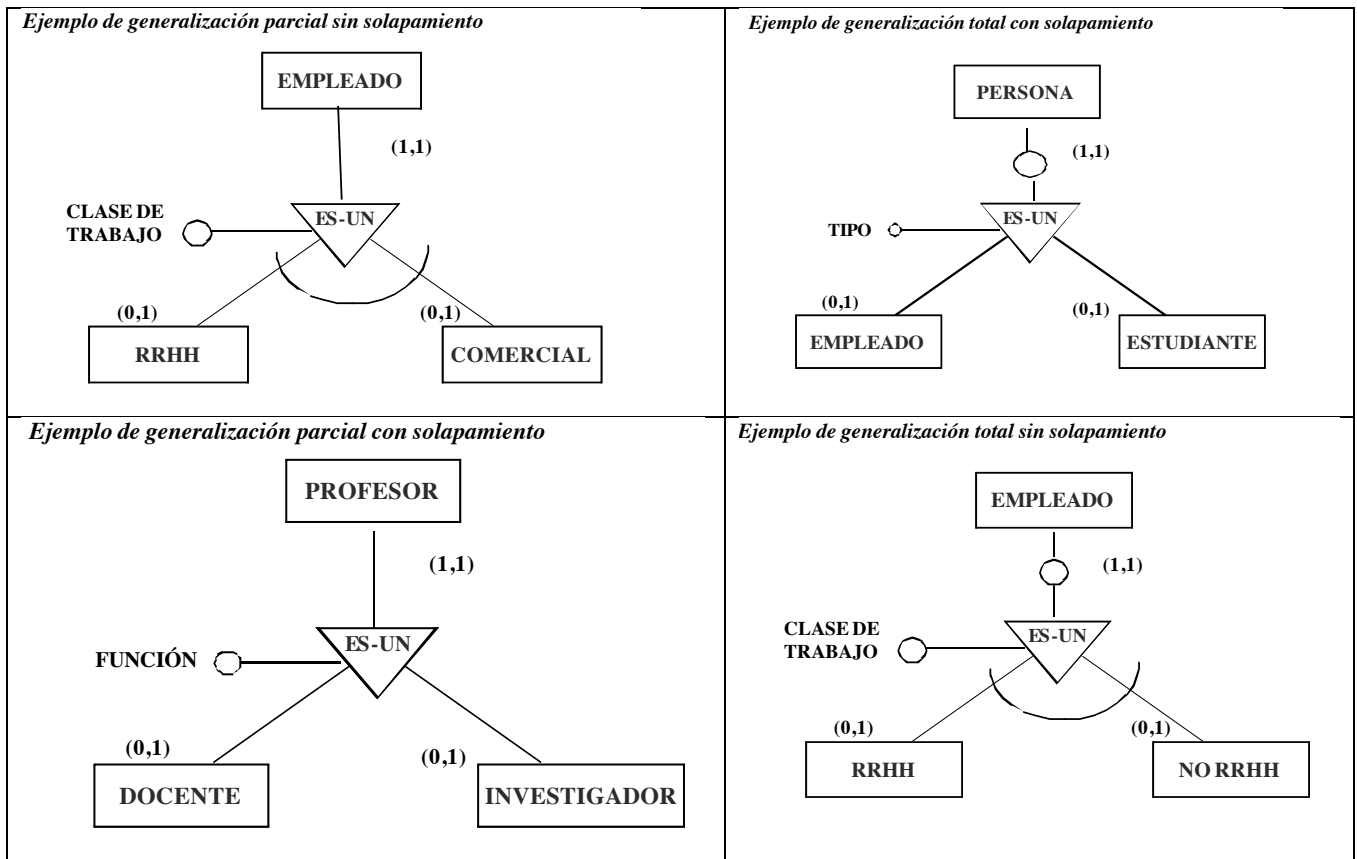
Este tipo de relación especial se representa a través de un triángulo invertido con la base paralela al rectángulo que representa al supertipo, y conectado a los subtipos. Esta relación tiene la característica de que toda ocurrencia de un subtipo es una ocurrencia del supertipo, aunque no sucede lo contrario, con lo que las cardinalidades serán siempre (1,1) en el supertipo, y (0,1) o (1,1) en el subtipo. El atributo del supertipo que actúa como discriminante se liga al triángulo a través de una elipse.

Una característica importante en estas relaciones es la herencia, puesto que cualquier atributo del supertipo pasa a ser un atributo de los subtipos.

Adicionalmente, existen dos tipos de elementos para la representación que se utilizan para indicar:

- **Disjunto/Solapado (Arco).** Indica la exclusividad de los subtipos, es decir, que una entidad del supertipo sólo puede ser de uno sólo de los subtipos.
- **Total/Parcial (Círculo).** Indica la obligatoriedad del supertipo de pertenecer a alguno de los subtipos.

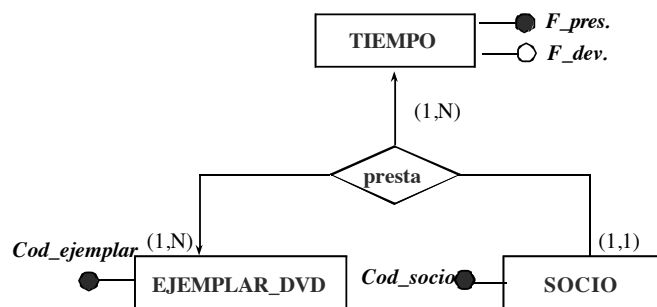
Tema 1 – Modelos de Datos y Modelo Entidad-Relación



c) Relaciones Ternarias (de grado 3)

Como ya se ha mencionado con anterioridad, lo habitual es encontrar relaciones binarias en los diagramas E/R. Sin embargo, esto no significa que no puedan existir relaciones de orden superior, aunque siempre es conveniente tratar de reducirlas (en la medida en que la semántica de la relación lo permita) a relaciones binarias. Vamos a analizar cómo se representan las relaciones ternarias, y cómo se deben interpretar las cardinalidades de estos tipos de relaciones.

Supongamos la relación ternaria mostrada en la figura inferior. En esta relación de ejemplo aparecen las entidades Profesor, Asignatura y Alumno. La relación se llama curso, y representa las ternas que se pueden dar durante un curso académico, relacionando los profesores que imparten ciertas asignaturas, con los alumnos matriculados en dichas asignaturas, y por ende, los profesores que tienen asignados ciertos alumnos.



Tema 1 – Modelos de Datos y Modelo Entidad-Relación

Es evidente que esta relación se podría descomponer en tres relaciones binarias, de modo que los profesores estuviesen relacionados con los alumnos a los que les imparten clase, los alumnos estuviesen relacionados con las asignaturas en las que están matriculados, y las asignaturas estuviesen relacionadas con los profesores que las imparten.

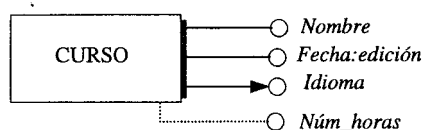
La información que se podría extraer de ambos diagramas E/R sería la misma, sólo que quizás se encontraría de forma más compacta en la representación tabular correspondiente al diagrama con la relación ternaria. No obstante, es mucho más fácil de interpretar el diagrama E/R con relaciones binarias

2.5 Claves. Tipos de Claves

Entre todos los atributos de un tipo de entidad han de existir uno o varios (simples y/o compuestos) que identifiquen unívocamente cada una de los ejemplares de ese tipo de entidad. Cada uno de estos conjuntos de atributos se denomina Identificador Candidato (IC) . Cuando un IC es compuesto, el número de los atributos que lo componen debe ser mínimo, en el sentido de que la eliminación de cualquiera de ellos le haría perder su carácter identificador. Luego todo IC debe cumplir la condición de ser unívoco y mínimo . Entre los IC se elige uno como Identificador Principal (IP) y el resto serán Identificadores Alternativos (IA). La representación gráfica de estos atributos queda reflejada a continuación: .



También podemos tener atributos opcionales (en cuyo caso los representaremos con líneas discontinuas) y atributos multivaluados (representados mediante una punta de flecha).



2.6 Normalización. Formas Normales

2.6.1 Concepto de Normalización

El proceso de Normalización se encarga según una serie de pasos o normas, que tras aplicar todas ellas, se obtienen datos agrupados en diferentes tablas, de tal forma que sea la estructura más óptima para su implementación, gestión y explotación desde diferentes futuras aplicaciones.

Una tabla se dice que está en una forma normal cuando satisface un conjunto de restricciones impuestas por dicha norma.

Tema 1 – Modelos de Datos y Modelo Entidad-Relación

El proceso de Normalización parte de las formas normales definidas por **E. F. Codd** (1970) creador de las Bases de Datos Relacionales.

Primeramente, Codd formuló las tres primeras formas normales (1FN, 2FN y 3FN), y posteriormente unas anomalías detectadas forzaron a crear una forma normal más completa que la tercera forma normal, que es la Forma Normal de Boyce y Codd (FNBC), después **Fagin** definió la cuarta y quinta forma normal (4FN y 5 FN).

La normalización se basa en que los datos sean independientes de las aplicaciones que los gestionan, y su objetivo es obtener el mayor número de tablas posibles, dejando en cada una de ellas los atributos imprescindibles para representar a la entidad (objeto).

El proceso de Normalización lo realiza el analista tras sucesivas reuniones con el usuario y toda la información que recibe de ellas se va documentando.

2.6.2 Primera Forma Normal (1FN)

Definición

Existen 2 definiciones:

Definición 1:

“Una tabla se dice que está en 1FN si y sólo si los valores que compone el atributo de una tupla son atómicos.”

Los valores de un atributo son atómicos cuando no aparecen valores repetidos, y por tanto tienen que ser elementales y únicos.

Definición 2:

“Una tabla se dice que está en 1FN si todos los atributos no clave, dependen funcionalmente de la clave”

Ejemplo: Tipos de Materiales de un ferretería.

TIPOS-MATERIALES

COD-MAT	DESCRIPCION	MEDIDAS
039	Tornillo	3.5-5-7
461	Broca	2-5
067	Arandela	2-5

Esta tabla no se encuentra en 1FN, puesto que el atributo Medidas tiene varios valores en diferentes tuplas, en vez de contener un único valor.

Algoritmo para pasar a 1FN:

Para que una tabla que no está en 1FN se pase a 1FN se procede:

Tema 1 – Modelos de Datos y Modelo Entidad-Relación

1º.- Se localizan los atributos que forman parte de la clave principal. Por ejemplo COD-MAT.

2º.- Se descompone la tabla realizando la Proyección:

2.1.- La clave con atributos que tienen valores únicos. Dicha tabla permanece con el nombre que la identifica. Por ejemplo:

TIPOS-MATERIALES

COD-MAT	DESCRIPCION
039	Tornillo
461	Broca
067	Arandela

2.2.- Otra tabla con la clave y los atributos que tiene valores múltiples (teniendo en cuenta que los valores múltiples se distribuirán cada uno en una tupla y por tanto en una fila existirá un solo valor elemental). La tabla que se genera tendrá por nombre mnemotécnico la abreviatura de los atributos que la forman. Por ejemplo:

T-MED-MAT

COD-MED-MAT	MEDIDA
03901	3.5
03902	5
03903	7
06701	2
06702	5

.....

.....

Dependencia Funcional (DF)

“Se dice que el atributo o conjunto de atributos B depende funcionalmente del atributo o conjunto de atributos A, y se representa como $A \rightarrow B$ ó $A \text{ DF } B$, si y sólo si cada valor de A se corresponde (a nivel conceptual) con un único valor de B.”

Por ejemplo: DNI \rightarrow Nombre, y si suponemos que no existen dos nombres iguales, entonces: Nombre \rightarrow DNI. Luego: DNI \leftrightarrow Nombre. Pero sin embargo, se cumple DNI \rightarrow Dirección, pero no: Dirección \rightarrow DNI.

Todo el proceso de Normalización está basado en distintas variantes de la dependencia funcional. Las Formas Normales que veremos más adelante se basan en la descomposición de tablas en función de las dependencias, y se basan en el Teorema (Pichat, 1978):

2.6.3 Segunda Forma Normal (2FN)

Dependencia Funcional Completa (DFC)

El atributo Y de la relación R tiene una dependencia completa del atributo X si tiene una dependencia funcional con X y no depende funcional de ningún subconjunto de X. Por

Tema 1 – Modelos de Datos y Modelo Entidad-Relación

ejemplo: DNI.Empresa \rightarrow Nombre, no es completa porque Nombre depende de DNI, la dependencia sería completa: DNI.Empresa \rightarrow Sueldo. Puesto que una empresa no tiene un único sueldo, y una persona (DNI) no tiene porqué tener un único sueldo, ya que puede estar trabajando en otra empresa.

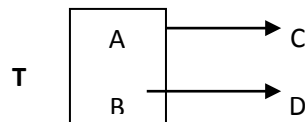
Definición de 2FN

Una relación está en 2FN si y sólo si cumple las condiciones:

- Se encuentra en 1FN.
- Todo atributo no primo (que no pertenece a la clave primaria) depende completamente de la clave primaria.

Esta forma normal sólo se considera si la Clave Primaria es compuesta, si no, ya está en 2FN.

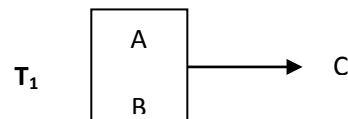
Por ejemplo: $A.B \rightarrow C$ y $B \rightarrow D$.



Esta tabla no está en 2FN, para convertirla en 2FN, realizamos la descomposición:

Algoritmo para pasar a 2FN:

1º.- Una tabla con la clave y todas sus dependencias completas con los atributos no primos afectados.

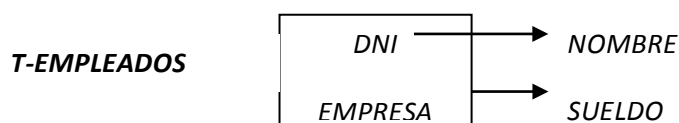


2º.- Otra tabla con la parte de la clave que tiene dependencias (no completas), junto con los atributos no primos afectados:

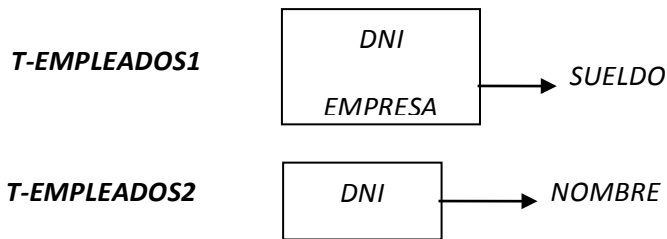


Por ejemplo: Representemos las relaciones que surgen de representar "Personas que trabajan en distintas empresas", los atributos son: DNI, Nombre, empresa y Sueldo.

Las dependencias funcionales son:



No está en 2FN, para pasarlo, descomponemos:



2.6.4 Tercera Forma Normal (3FN)

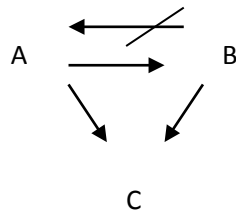
Una Relación está en 3FN si y sólo si se cumplen las condiciones:

- a) Se encuentra en 2FN
- b) No existen atributos no primos que sean transitivamente dependientes de cada posible clave de la tabla.

Esto quiere decir que un atributo no primo (que no forma parte de la clave primaria) sólo se debe conocer a través de la clave principal de la tabla y no por medio de otro atributo no primo.

Dependencia Funcional Transitiva:

Sean tres subconjuntos distintos de atributos A, B y C pertenecientes a una tabla T, de tal modo que se cumplen las condiciones: $A \rightarrow B$ y $B \rightarrow C$. se dice que C tiene una dependencia funcional transitiva con A, o que es transitivamente dependiente si se cumple $B \rightarrow C$, gráficamente sería:



Algoritmo para pasar a 3FN

1º.- Sea una Dependencia Funcional $X \rightarrow Y$ donde X e Y son disjuntas y no son clave ni forman parte de ella.

2º.- Se obtiene las proyecciones:

2.1.- $T_1 = P(X, Y)$

2.2.- $T_2 = P(U - Y)$

Siendo U todos los atributos de la tabla.

Si la tabla T_2 sigue sin estar en 3FN se repite el algoritmo para T_2 .

3 CONSTRUCCIÓN MODELO LÓGICO DE DATOS

3.1 Introducción

En el diseño lógico se deben de coordinar exigencias casi siempre encontradas, como son eliminar redundancias, conseguir la máxima simplicidad, y evitar cargas suplementarias de programación, obteniendo una estructura lógica adecuada que venga a establecer el debido equilibrio entre las exigencias de los usuarios y la eficiencia.

Pero básicamente, el proceso de diseño lógico debe utilizar como entrada un modelo de datos conceptual, el cual se deberá convertir al modelo de la base de datos para el que se está diseñando, esto es, el **diseño lógico consiste en una transformación de modelos**.

Adicionalmente, será necesario comprobar que el modelo lógico resultante garantiza las propiedades de eliminación de redundancias, fácil accesibilidad a los datos, etc. Para ello, en el caso de bases de datos relacionales, recurriremos a la teoría de la normalización, que nos permitirá obtener las mejores relaciones posibles para un esquema lógico.

3.2 Transformación de diagramas E/R en relaciones

El paso de un esquema en el modelo E/R al relacional está basado en los tres principios siguientes:

- a) Todo tipo de entidad se convierte en una relación.
- b) Todo tipo de relación N:M se transforma en una relación.
- c) Todo tipo de relación 1:N se traduce en el fenómeno de propagación de clave o se crea una nueva relación.

En el proceso de transformación de esquemas se **pierde semántica**, pero esto no va a afectar para nada la integridad de la base de datos, ya que se definirán, por ejemplo, restricciones de integridad referencial para asegurar la conservación de la misma.

3.2.1 Transformación de Entidades

Cada tipo de entidad se debe convertir a una relación, es decir, será necesario crear una tabla para cada entidad que aparezca en el diagrama E/R. Esto, sin embargo, tiene alguna restricción para el caso particular en que las relaciones entre entidades sea del tipo 1:1.

3.2.2 Transformación de Atributos de Entidades

Cada atributo de una entidad se debe transformar en una columna en la relación a la que ha dado lugar la entidad. Puesto que hay diferentes tipos de atributos, hay que indicar cómo se deben definir cada uno de ellos:

El o los atributos principales de una entidad (es decir, los que actúan como identificador único para los elementos de la relación, subrayados en el esquema) pasan a ser la clave primaria de la relación. Se debe especificar que no son nulos.

El resto de atributos pasan a ser columnas de la tabla, pudiendo tomar valores nulos, a no ser que se indique lo contrario.

En el caso de tener un atributo multivaluado, deberemos de eliminar este atributo construyendo una relación 1:N a partir de dicho atributo.

3.2.3 Transformación de Relaciones

Dependiendo del tipo de relación (cardinalidad) de que se trate, existen diversas formas de transformarlas:

- **Relaciones N:M.** Se transforman en una relación que tendrá como clave primaria la concatenación de los atributos principales de cada una de las entidades que relacionan. Estos atributos deben ser clave ajena respecto a cada una de las tablas donde ese atributo es clave primaria. Habrá que considerar lo que ocurre en los casos en los que se borre o modifique la clave primaria referenciada. Será necesario crear las restricciones de cardinalidad adecuadas a través de la sentencia CHECK de SQL.
- **Relaciones 1:N.** Existen dos soluciones para transformarlas. Habrá que considerar en ambos casos las referencias ajenas, y las actuaciones en caso de borrado y modificación.
 - a) Propagar el atributo principal de la entidad que tiene cardinalidad máxima 1 a la que tiene N, y hacer desaparecer la relación como tal.
 - b) Transformarla en una relación como si fuese una de tipo N:M. Esta acción sólo es recomendable en el caso de que la correspondencia 1, se debiera a una cardinalidad (0,1)
- **Relaciones 1:1.** Puesto que se trata de una particularización de cualquiera de los dos casos anteriores, se puede del mismo modo aplicar cualquiera de las dos reglas definidas. No obstante, es recomendable seguir las siguientes reglas:
 - Si la relación es (0,1):(0,1), es mejor crear una relación para evitar el tener muchos nulos como propagación de alguna de las claves a la otra relación (si se prevé que puede haber muchos nulos)
 - Si la relación es (0,1)(1,1), hay que propagar la clave de la entidad (1,1) a la (0,1)

- Si la relación es (1,1):(1,1) la propagación es indiferente, y se hará atendiendo a los criterios de frecuencia de acceso (consulta, modificación, inserción, etc.) a cada una de las tablas en cuestión.

3.2.4 Transformación de atributos de relaciones

Si la relación se transforma en una tabla, todos sus atributos pasan a ser columnas de la tabla.

En este caso habrá que analizar si dichos atributos de la relación deberá ser incluido como parte de la clave primaria.

Si la relación se pierde, porque se propaga su clave, deberemos de propagar también todos los atributos de dicha relación como campos simples.
