

UNIDAD DIDÁCTICA 5

NORMALIZACIÓN DE RELACIONES

CONTENIDO

1. Normalización de los Modelos Relacionales
 - 1.1. Primera Forma Normal (1FN)
 - 1.2. Dependencias Funcionales
 - 1.2.1. Segunda Forma Normal (2FN)
 - 1.2.2. Tercera Forma Normal (3FN)
 - 1.2.3. Forma Normal de Boyce-Codd (FNBC)
 - 1.3. Atributos compuestos
 - 1.4. Atributos multivaluados

OBJETIVOS

- Utilizar herramientas gráficas para representar el diseño lógico.
- Aplicar las reglas de normalización.

INTRODUCCIÓN

Habitualmente el diseño de una base de datos termina en el paso del modelo **Entidad – Relación** al modelo **Relacional**. No obstante, siempre que se diseña un sistema, no solo una base de datos, sino también cualquier tipo de solución informática, hay que medir la calidad de la misma, y si no cumple determinados criterios de calidad, hay que realizar, de forma iterativa, sucesivos refinamientos en el diseño, para alcanzar la calidad deseada. Uno de los parámetros que mide la calidad de una base de datos es la forma normal en la que se encuentra el diseño. Esta forma normal puede alcanzarse cumpliendo ciertas restricciones que impone cada forma normal al conjunto de atributos de un diseño. El proceso de obligar a los atributos de un diseño a cumplir ciertas formas normales se llama **normalización**.

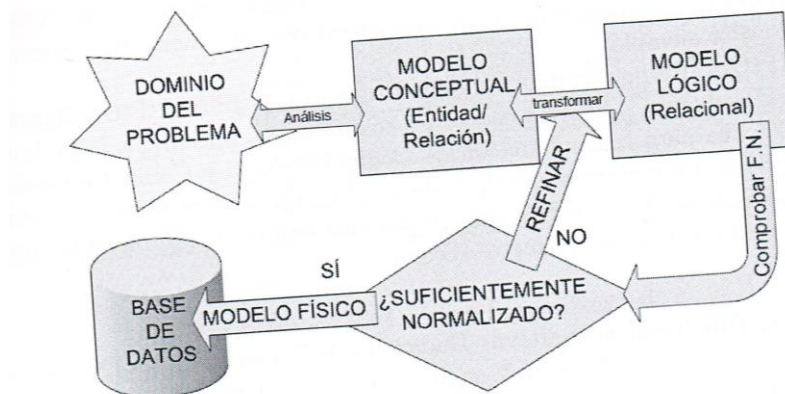


Fig.1: Refinamiento de un diseño de base de datos

Las formas normales pretenden alcanzar dos objetivos:

1. Almacenar en la base de datos cada hecho solo una vez, es decir, evitar la redundancia de datos. De esta forma se reduce el espacio de almacenamiento.
2. Que los hechos distintos se almacenen en sitios distintos. Esto evita ciertas anomalías a la hora de operar con los datos.

A medida que se alcanza una forma normal más avanzada se cumplen mejor estos objetivos.

Ejemplo de problemas derivados de un diseño inadecuado. Suponemos la Tabla ESCRIBE que almacena datos sobre los libros (CodLibro, Título, Editorial y Año) y sobre los autores que los han escrito (Autor y Nacionalidad)

ESCRIBE					
AUTOR	NACIONALIDAD	CODLIBRO	TITULO	EDITORIAL	ANHO
Adoración de Miguel	Española	23433	Fundamentos de BBDD	RAMA	1999
Mario Piattini	Italiana	23454	Fundamentos de BBDD	RAMA	1999
Edgar D'Andrea	Norteamericana	34567	MySQL 5	InfoBook's	2009
Irene Luque	Española	89987	Bases de Datos	RAMA	2005
Miguel Ángel Gómez	Española	89988	Bases de Datos	RAMA	2005
Enrique López	Española	89989	Bases de Datos	RAMA	2005

Mario Piattini	Italiana	14587	Auditoría de Tecnologías y Sistemas de la Información	RAMA	2008
----------------	----------	-------	---	------	------

En esta tabla los principales problemas que observamos se derivan de la gran cantidad de redundancias que presenta. Por ejemplo, la nacionalidad del autor se repite por cada libro que haya escrito; y algo parecido, sucede cuando un libro tiene más de un autor, con la editorial y el año de publicación.

Esta redundancia produce a su vez:

- **Anomalías de inserción**, ya que dar de alta un libro obliga a insertar en la base de datos tantas tuplas como autores tenga el libro.
- **Anomalías de modificación**. Ya que cambiar la editorial de un libro obliga a modificar todas las tuplas que corresponden a ese libro.
- **Anomalía de borrado**, ya que el borrado de un libro obliga a borrar varias tuplas, tantas como autores tenga ese libro y viceversa, el borrado de un autor nos lleva a borrar tantas tuplas como libros ha escrito ese autor.

Por lo tanto, la actualización (altas, bajas, modificaciones) de un solo libro o de un autor nos puede obligar a actualizar más de una tupla, dejando la integridad de la base de datos en las manos del usuario. Lo que supone una falta de eficiencia en las actualizaciones.

Además de estas anomalías de inserción, borrado y modificación, existen otros problemas adicionales, como la **imposibilidad de almacenar ciertos hechos** o la **desaparición de información** que desearíamos mantener en la base de datos.

Por ejemplo, si se quisiera incluir información sobre un autor del que no existiera ningún libro en la base de datos, no sería posible, ya que el atributo CODLIBRO forma parte de la clave primaria de la tabla; ni tampoco podríamos introducir obras. (No se permiten nulos en la clave primaria). Por otro lado, al dar de baja un libro, se pierden también los datos de los autores, en el caso de que tuviesen un único libro en la base de datos, y viceversa, si borramos un autor desaparecen de la base de datos los libros escritos por él, a no ser que el libro tuviera más de un autor.

Esta relación origina tantos problemas debido a que atenta contra un principio básico del diseño:

“Hechos distintos se deben almacenar en objetos distintos.”

En nuestro caso, en tablas distintas, con lo que nos evitaríamos redundancias y por lo tanto, los demás problemas descritos.

Con un diseño riguroso no tendríamos una tabla de ese tipo. Si primero se realiza un buen diseño conceptual en el modelo E/R, seguido de una cuidadosa transformación al modelo relacional, se evitarían en gran parte estas anomalías, obteniéndose en general un esquema exento de errores. Sin embargo, ante las dudas respecto a si un esquema relacional es o no correcto, es preferible aplicar siempre al esquema un método formal de análisis que nos ayude a detectar los posibles errores y llegar a otro esquema en el que aseguremos el cumplimiento de ciertos requisitos, a este método formal se le llama la **“teoría de la normalización”**:

La teoría de la normalización evita las redundancias y las anomalías de actualización, obteniendo tablas más estructuradas que no presentan los problemas anteriores.

En lugar de la tabla anterior, se podría haber diseñado el siguiente esquema relacional:

LIBRO(CodLibro, Titulo, Editorial, Año)

AUTOR(Nombre, Nacionalidad)

(ESCRIBE(CodLibro, Nombre)

Donde al seguir el principio anterior, se han separado hechos distintos en tablas distintas, de forma que cada una de estas tablas recoja un hecho determinado y concreto del mundo real con sus correspondientes atributos.

La teoría de la normalización, se puede definir como una técnica formal para organizar datos, que nos ayuda a determinar qué es lo que está equivocado en un diseño y nos enseña la manera de corregirlo.

Normalización de los Modelos Relacionales

El modelo relacional está basado en los principios de la teoría de conjuntos. Aunque una tabla representa a un conjunto, no todos los conjuntos pueden ser considerados como esquemas relacionales. Para ser considerado un esquema relacional debe cumplir los siguientes objetivos:

- No existencia de redundancias superfluas, reduciendo el espacio necesario para el almacenamiento y reduciendo posibles problemas de integridad en la información almacenada en la base de datos.
- Aumentar el desempeño de las operaciones de actualización de la base de datos.
- Representar de forma coherente los objetos y relaciones existentes en el dominio del problema y cuya información es almacenada en la base de datos.
- Aumentar el desempeño y garantizar la fiabilidad de las consultas sobre la información de la base de datos.

Para cumplir estos objetivos, las tablas que forman un esquema relacional deben cumplir una serie de reglas, denominadas **Reglas de Normalización de Relaciones** y a la teoría en la que se basan se le denomina **Teoría de la Normalización de Relaciones**.

Se dice que una relación está en una determinada forma normal si satisface un cierto conjunto específico de restricciones impuestas por la regla de normalización correspondiente.

La sucesiva aplicación de las reglas de normalización va a dar lugar a la generación de un número mayor de relaciones o tablas que formen parte del esquema relacional, y restringe el número de relaciones que las cumplen.

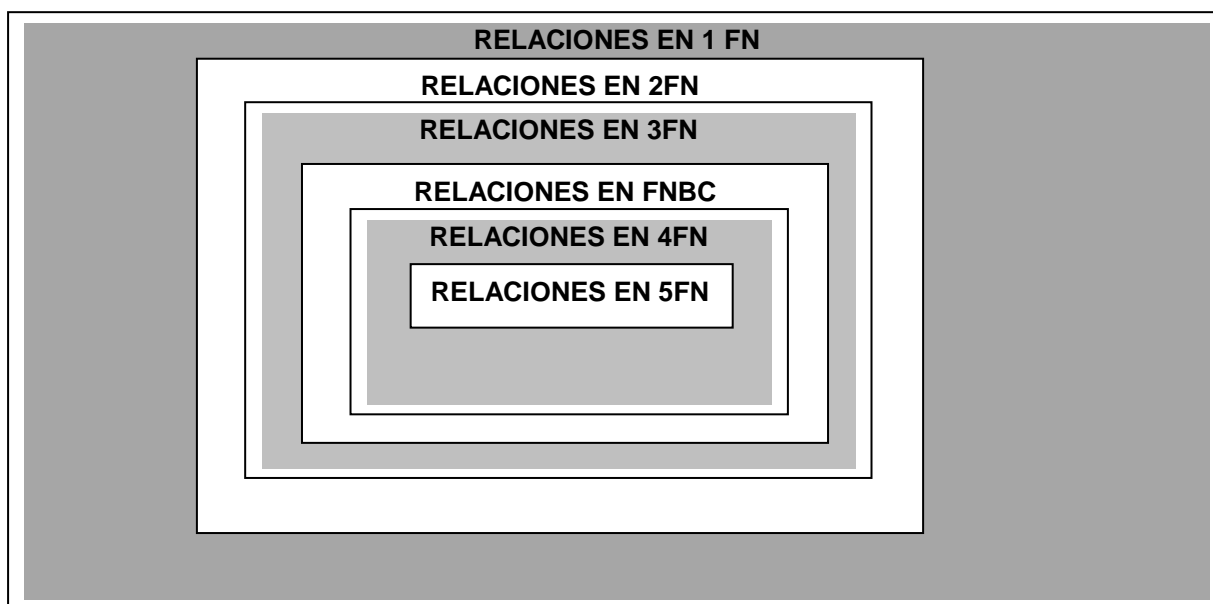
Por regla general, se dice que un esquema relacional es consistente si las relaciones satisfacen al menos la forma normal de Boyce-Codd (FNBC).

Noción intuitiva de las formas normales

La teoría de la normalización se centra en las formas normales. Se dice que un esquema relacional está en una determinada forma normal si satisface un conjunto de restricciones.

La primera forma normal (1FN) fue introducida por Codd en su primer trabajo, es una restricción inherente al modelo relacional, por lo que su cumplimiento es obligatorio. Consiste en la prohibición de que en una tabla existan grupos repetitivos, es decir, que un atributo pueda tomar más de un valor de su dominio.

Codd definió también las 2FN y la 3FN, más adelante otros autores propusieron nuevas formas normales la Forma Boyce-Codd (FNBC) y la cuarta (4FN) y la quinta (5FN) debidas a FAGIN.



De todas las relaciones que se encuentran en la forma 1FN, sólo algunas se encuentran en la forma 2FN, y de éstas una parte en la forma 3FN y así sucesivamente; es decir, que la forma 2FN impone más restricciones que la 1FN, siendo la 5FN la forma que impone más restricciones, después la 4FN y así hasta la 1FN.

La teoría de la normalización se basa en ciertos criterios definidos sobre los atributos de una tabla, llamados dependencias, existen distintos tipos de dependencias:

Dependencias Funcionales	1FN
	2FN
	3FN
	FNBC
Dependencias Multivaluadas	4FN
Dependencias de Proyección-Combinación	

La teoría formal de la Normalización requiere una base matemática, por ello se va a explicar de forma más intuitiva.

Enfoque Intuitivo

Primera Forma Normal 1FN: En una tabla no puede haber atributos multivaluados, como es una restricción inherente del modelo relacional, en general, la mayoría de las tablas se encuentran en 1FN.

Aplicar la primera forma normal es muy simple, bastará con dividir cada columna no atómica en tantas columnas atómicas como sea necesario. Por ejemplo, si tenemos una relación que contiene la información de una agenda de amigos con este *esquema*:

Agenda (Nombre, email)

El nombre, normalmente, estará compuesto por el tratamiento (señor, señora, don, doña, excelencia, alteza, señoría, etc.), un nombre de pila y los apellidos. Podríamos considerar el nombre como un dato atómico, pero puede interesarnos separar algunas de las partes que lo componen.

¿Y qué pasa con la dirección de correo electrónico? También podemos considerar que es un valor no atómico, la parte a la izquierda del símbolo @ es el usuario, y a la derecha el dominio. De nuevo, dependiendo de las necesidades del cliente o del uso de los datos, podemos estar interesados en dividir este campo en dos, o incluso en tres partes (puede interesar separar la parte a la derecha del punto en el dominio).

Tanto en esta forma normal, como en las próximas que veremos, es importante no llevar el proceso de normalización demasiado lejos. Se trata de facilitar el trabajo y evitar problemas de redundancia e integridad, y no de lo contrario. Debemos considerar las ventajas o necesidades de aplicar cada norma en cada caso, y no excedernos por intentar aplicar las normas demasiado al pie de la letra.

El esquema de la relación puede quedar como sigue:

Agenda (Nombre_Tratamiento, Nombre_Pila, Nombre_Apellidos, email)

Otro caso frecuente de relaciones que no cumplen 1FN es cuando existen atributos multivaluados, si todos los valores se agrupan en un único atributo:

Libros (Titulo, autores, fecha, editorial)

Hemos previsto, que un libro puede tener varios autores. No es que sea muy frecuente pero sucede, y más con libros técnicos y libros de texto.

Sin embargo, esta relación no es 1FN, ya que en la columna de autores sólo debe existir un valor del dominio, por lo tanto debemos convertir ese atributo en uno multivaluado:

Libros			
Titulo	autor	fecha	editorial
Diseño de Bases de Datos	Adoración de Miguel, Paloma Martínez, Elena Castro, Jose M ^a Cavero Dolores Cuadra, Ana M ^a Iglesias, Carlos Nieto	12/10/2003	RAMA

Programación en lenguajes Estructurados	Félix García, Jesús Carretero, José Manuel Pérez Alejandro Calderón Javier Fernández	23/09/2009	Paraninfo
MySQL 5	Edgar D'Andrea		Info'Books

Libros			
Titulo	autor	fecha	editorial
Diseño de Bases de Datos	Jose M ^a Cavero	12/10/2003	RAMA
Diseño de Bases de Datos	Paloma Martínez,	12/10/2003	RAMA
Diseño de Bases de Datos	Elena Castro	12/10/2003	RAMA
Diseño de Bases de Datos	Adoración de Miguel,	12/10/2003	RAMA
Diseño de Bases de Datos	Dolores Cuadra	12/10/2003	RAMA
Diseño de Bases de Datos	Ana M ^a Iglesias	12/10/2003	RAMA
Diseño de Bases de Datos	Carlos Nieto	12/10/2003	RAMA
Programación en lenguajes Estructurados	Félix García	23/09/2009	Paraninfo
Programación en lenguajes Estructurados	Jesús Carretero,	23/09/2009	Paraninfo
Programación en lenguajes Estructurados	José Manuel Pérez	23/09/2009	Paraninfo
Programación en lenguajes Estructurados	Alejandro Calderón	23/09/2009	Paraninfo
Programación en lenguajes Estructurados	Javier Fernández	23/09/2009	Paraninfo
MySQL 5	Edgar D'Andrea	01/01/2010	Info'Books

Segunda Forma Normal 2FN se dice que una tabla está en 2FN, si se encuentra en la 1FN y además todos los atributos que no forman parte de ninguna clave candidata suministran información acerca de la clave completa, no de una parte de la clave.

Ejemplo, suponemos la siguiente tabla diseñada para guardar los libros prestados a los socios de una biblioteca.

PRESTAMO (NumSocio, DNISocio, CodLibro, FechaPrestamo, Editorial, Pais)

Donde las posibles claves candidatas son: (NumSocio, CodLibro) y (DNISocio, CodLibro). Se puede observar que hay ciertos atributos en la tabla que no forman parte de ninguna clave candidata que nos dan información acerca del libro y no tienen nada que ver con el socio, por lo que no son información acerca de la clave completa sino únicamente de una parte de la clave CodLibro, por lo que la tabla PRESTAMO, no está en 2FN.

Claves Candidatas	Atributos
NumSocio, CodLibro	FechaPrestamo (depende de la clave candidata completa)
DNISocio, CodLibros	FechaPrestamo (depende de la clave candidata completa)

Si analizamos el resto de los atributos de la tabla

Editorial	no depende de la clave completa, sólo da información del libro depende de CodLibro
Pais	no depende de la clave completa, sólo da información del libro depende de CodLibro

La solución al problema es crear una nueva tabla con los atributos que no dependen de la clave completa.

PRESTAMO (NumSocio, DNISocio, CodLibro, FechaPrestamo)

LIBRO (CodLibro, Editorial, Pais)

Si analizamos estas dos nuevas tablas, vemos que en la tabla PRESTAMO hay dos claves candidatas (NumSocio, CodLibro) y (DNISocio, CodLibro) y un único atributo (FechaPrestamo) que no forma parte de la clave candidata, pero suministra información acerca de la totalidad de ambas claves candidatas. En LIBRO la clave es CodLibro y los dos atributos que no son clave también suministran información acerca de la clave completa.

Por lo tanto, ambas tablas están en 2FN.

Regla: *Toda tabla cuya clave está formada por un único atributo está al menos en 2FN.*

Tercera Forma Normal 3FN. Una tabla se encuentra en 3FN si además de estar en la 2FN y por lo tanto en 1FN, se cumple que los atributos que no forman parte de ninguna clave candidata facilitan información sólo acerca de la(s) clave(s) y no acerca de otros atributos.

Ejemplo, suponemos la siguiente tabla

PRESTAMO (NumSocio, DNISocio, CodLibro, FechaPrestamo)

En esta tabla el atributo FechaPrestamo sólo facilita información acerca de las claves, ya que no existe ningún otro atributo no clave, la tabla está en 3FN. Si analizamos la tabla LIBRO.

LIBRO (CodLibro, Editorial, Pais)

Se puede comprobar que el atributo Pais facilita información acerca de la Editorial y no sobre el campo clave, por lo que se puede afirmar que LIBRO no está en la 3FN.

Para resolver el problema, redundancias, inconsistencias, etc, que se originan en la tabla, conviene descomponerlas en:

LIBRO (CodLibro, Editorial)

EDITORIALES (Editorial, Pais)

Si volvemos a analizar estas tablas, ya están en 3FN, ya que todo atributo no clave sólo facilita información acerca de la clave.

Resumen para que una relación esté en 3FN, ha de estar en 1FN y además todos sus atributos que no forman parte de ninguna clave candidata, "*deben ser información referida a la clave, la clave completa y nada más que la clave*", KENT (1983).

Al descomponer la relación inicial PRESTAMO, que no estaba en 2FN, en tres relaciones que están en 3FN, se evitan muchas de las anomalías que se producen, pero todavía pueden existir problemas y para evitarlos se definieron el resto de las formas normales.

Dependencias Funcionales

La normalización de tablas se basa en el concepto de dependencias, por lo que también se conoce como teoría de las dependencias, que se centra en el estudio de las dependencias que presenta cada atributo de una tabla con respecto al resto de atributos de la misma tabla, es decir, las dependencias nos muestran las relaciones existentes entre los atributos.

Las dependencias funcionales son un tipo especial de dependencias, el más extendido en la práctica, en el que se basan las primeras formas normales.

Enunciado de la teoría de las dependencias: sea el esquema de Tabla T definido sobre el conjunto de atributos A y sean X a Y subconjuntos de A llamados *descriptores*. Se dice que Y **depende funcionalmente** de X , o lo que es lo mismo, que X determina a Y si, y sólo si, cada valor de X tiene asociado en todo momento un único valor de Y . La dependencia funcional se representa:

$X \rightarrow Y$

Se llama determinante o implicante al descriptor que se encuentra a la izquierda del símbolo de implicación, e implicado al descriptor que se encuentra a la derecha.

Ejemplo: LIBRO(CodLibro, Titulo, Idioma,...)

Podemos decir que el código de un libro determina el título del mismo.

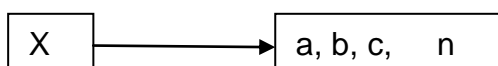
CodLibro \rightarrow Titulo

El código del libro (CodLibro) es el implicante o determinante en la dependencia y Titulo es el implicado. Esta dependencia también nos dice que el título es una información acerca del libro, ya que una dependencia funcional se puede interpretar diciendo que el implicado es un hecho (o información) acerca del implicante. Esto no quiere decir, que conocido el código de un libro podamos deducir el título del libro. Sino que dadas dos filas que tengan el mismo CodLibro el título será igual en ambas.

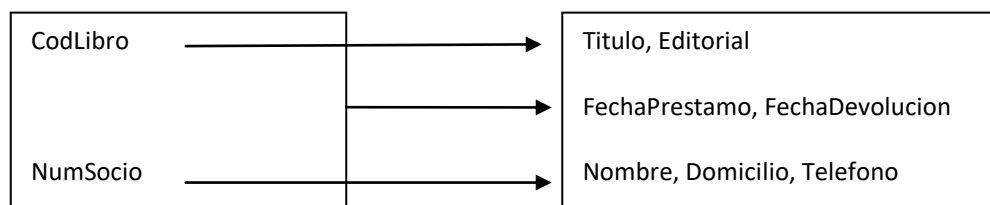
Es decir, para un esquema T tenemos la dependencia funcional $X \rightarrow Y$, dado el valor de X no podemos, en general, conocer el valor de Y ; solamente se puede afirmar, que dadas dos tuplas o filas que tengan el mismo valor de X , el valor de Y será el mismo en ambas.

Para representar las dependencias funcionales se utilizan los grafos o diagramas de dependencias funcionales y la representación textual.

- Representación gráfica, grafos, mediante el cual se representa un conjunto de atributos y las dependencias funcionales existentes entre ellos. En el grafo aparecen los nombres de los atributos unidos por flechas, las cuales indican las dependencias funcionales y parten del implicante hacia el implicado. Cuando el implicante no es el único atributo, es decir, se trata de un implicante compuesto, los atributos que lo componen se encierran en un recuadro y la flecha parte de él, no de cada atributo.



Ejemplo: PRESTAMO (CodLibro, Titulo, Editorial, NumSocio, Nombre, Domicilio, Telefono, FechaPrestamo, FechaDevolución)



En el grafo se visualizan las siguientes dependencias: CodLibro determina funcionalmente el Título del libro y la Editorial, como nos indica la flecha; de igual forma NumSocio determina el Nombre, Domicilio y Telefono del socio (suponemos sólo uno); mientras que ambos atributos CodLibro y NumSocio determinan FechaPrestamo y FechaDevolución, por lo que la flecha sale del recuadro.

Representación textual

$X \rightarrow (a, b, c, \dots n)$

CodLibro \rightarrow (Titulo, Editorial)

CodLibro, NumSocio \rightarrow (FechaPrestamo, FechaDevolucion)

NumSocio \rightarrow (Nombre, Domicilio, Telefono)

a) Dependencia Funcional Plena o Completa

Sea un descriptor compuesto X: $X(X_1, X_2)$

Se dice que Y tiene dependencia funcional completa o plena de X si depende funcionalmente de X pero no depende de ningún subconjunto del mismo.

$X \rightarrow Y$

$X_1 \nrightarrow Y$

$X_2 \nrightarrow Y$

Y se representa $X \Rightarrow Y$

Ejemplo: Suponemos la relación:

PRESTAMO (CodLibro, Titulo, Editorial, NumSocio, Nombre, Domicilio, Telefono, FechaPrestamo, FechaDevolución)

Cuyas dependencias funcionales son:

CodLibro \rightarrow (Titulo, Editorial)

CodLibro, NumSocio \rightarrow (FechaPrestamo, FechaDevolucion)

NumSocio \rightarrow (Nombre, Domicilio, Telefono)

La dependencia funcional:

CodLibro, NumSocio \rightarrow (FechaPrestamo)

Indica que, dado un determinado código de libro (CodLibro) y un número de socio (NumSocio), existe una única fecha de préstamos (suponemos que un mismo libro no se presta a un mismo socio más de una vez. Ni CodLibro, ni NumSocio implican por sí solos, la fecha del préstamo, ya que tanto un libro se puede prestar en varias fechas como un socio puede recibir libros prestados en varias fechas. Por tanto, la dependencia funcional anterior es completa. Lo que se representa.

CodLibro, NumSocio \Rightarrow (FechaPrestamo)

Ya que: CodLibro \nrightarrow (FechaPrestamo)

NumSocio \nrightarrow (FechaPrestamo)

Lo que se puede interpretar como que FechaPrestamo constituye una información sobre el libro y el socio, pero esta información no atañe a un libro o a un socio por separado.

Por el contrario, la dependencia CodLibro, NumSocio \Rightarrow (Editorial)

No es funcional, ya que:

CodLibro \rightarrow Editorial

Se dice que en esa dependencia, NumSocio es un atributo redundante o ajeno a la dependencia, también se le llama extraño.

b) Dependencia Funcional Transitiva

Sea la Relación o Tabla $R(X, Y, Z)$

En la que existen las siguientes dependencias funcionales:

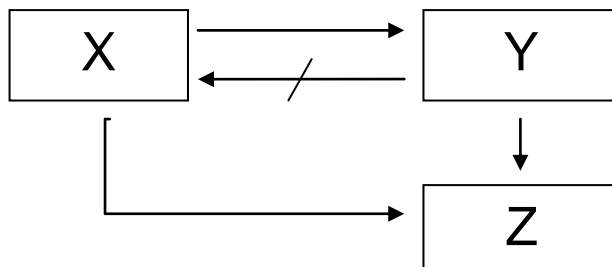
$$X \rightarrow Y \quad Y \rightarrow Z \quad Y \not\rightarrow X$$

Se dice que entonces Z tiene una dependencia transitiva respecto de X a través de Y, que se representa

$$X \twoheadrightarrow Z$$

Si además $Z \not\rightarrow Y$, se dice que la dependencia transitiva es estricta

Se representaría:



Ejemplo: Si consideramos la relación LIBROS(CodLibro, Editorial, Pais), en donde tenemos para libro su código, la editorial que lo publica y el país de la editorial, suponemos que la editorial tiene su sede en un único país, se tendrán las siguientes dependencias:

$\text{CodLibro} \rightarrow \text{Editorial}$

$\text{Editorial} \rightarrow \text{Pais}$

$\text{CodLibro} \twoheadrightarrow \text{Pais}$

Además, $\text{Editorial} \not\rightarrow \text{CodLibro}$ (ya que una editorial puede publicar varios libros), la dependencia funcional entre CodLibro y Pais es una dependencia transitiva a través de Editorial, se representa

$\text{CodLibro} \twoheadrightarrow \text{Editorial}$

Lo que se puede interpretar intuitivamente como que Pais proporciona información sobre el libro, pero indirectamente, ya que constituye una información sobre la editorial y ésta a su vez sobre el libro.

Además, como $\text{Pais} \not\rightarrow \text{Editorial}$, la dependencia transitiva es estricta.

Si la relación a analizar fuese:

LIBROS(CodLibro, Titulo, Editorial)

Tendríamos las siguientes dependencias:

$\text{CodLibro} \rightarrow \text{Titulo}$

Titulo → CodLibro

CodLibro → Editorial

Titulo → Editorial

Ninguna de las dependencias CodLibro → Editorial, Titulo → Editorial son transitivas, al ser los atributos CodLibro y título equivalentes, ambos se implican mutuamente.

Primera Forma Normal 1FN

Una relación o tabla R satisface la primera forma normal (1FN) si, y sólo si, todos los dominios subyacentes de la relación R contienen valores atómicos.

Atómico significa "indivisible", es decir, cada atributo debe contener un único valor del dominio. Los atributos, en cada tabla de una base de datos 1FN, no pueden tener listas o arrays de valores, ya sean del mismo dominio o de dominios diferentes.

La aplicación de esta regla es fácil y directa para cualquier relación, y este proceso se realiza de forma automática en el proceso de análisis del dominio del problema. Simplemente consiste en descomponer aquellas tuplas en las que los atributos tengan más de un valor en tantas tuplas como valores estén presentes. Ésta es una restricción innata del propio modelo relacional.

El que una relación esté en 1FN no es condición suficiente, aunque si necesaria, para garantizar la consistencia del esquema relacional.

Usuarios				
IdUsuario	Nombre	Empresa	DirecEmpresa	url
1	Juan	ABC	Eibar	Abc.com Juan.com
2	Ana	XYZ	Vigo	Xyz.com
3	María	MPR	Madrid	Mpr.com Maria.com

Solucionamos el problema del atributo url y la tabla o relación se transforma en:

Usuarios				
IdUsuario	Nombre	Empresa	DirecEmpresa	url
1	Juan	ABC	Eibar	Abc.com
1	Juan	ABC	Eibar	Juan.com
2	Ana	XYZ	Vigo	Xyz.com
3	María	MPR	Madrid	Mpr.com
3	María	MPR	Madrid	Maria.com

Ahora diremos que nuestra tabla está en 1FN. Hemos solucionado el problema de la limitación del campo url. Pero sin embargo vemos otros problemas. Cada vez que introducimos un nuevo registro en la **tabla usuarios**, tenemos que duplicar el nombre de la empresa y del usuario. No sólo nuestra BD crecerá muchísimo, sino que será muy fácil que la BD se corrompa si escribimos mal alguno de los datos redundantes.

Segunda Forma Normal 2FN

Una relación R satisface la segunda forma normal (2FN) si, y sólo si, satisface la 1FN y cada atributo de la relación depende funcionalmente de forma completa de la clave primaria de esa relación.

Ejemplo: Analizamos la Relación Matricula

MATRICULA(dni, asignatura, apellidos, nombre, nota, curso, aula, lugar)

El atributo asignatura representa el código de las asignaturas de las está matriculado cada uno de los alumnos.

El atributo **aula** representa las aula en las que se imparten las asignaturas.

El atributo **lugar** representa las ciudades en las que se imparten las asignaturas.

El atributo **curso** representa el curso en el que se imparte una asignatura.

Si analizamos las dependencias podemos observar que existe una dependencia funcional entre los atributos asignatura y curso, que representa que en un curso se pueden impartir varias asignaturas, y una asignatura está asignada a un curso. Asignatura → Curso Por lo que esta relación no está en la 2FN, puesto que existe una dependencia funcional no completa entre matricula y curso, es decir, curso no depende de la clave completa.

La relación matricula presenta los siguientes inconvenientes debido a que no cumple 2FN:

Inserción de tuplas, no se puede conocer las asignaturas que se imparten en un curso hasta que no exista algún alumno matriculado en esas asignaturas. Por ejemplo, si la asignatura '4' se imparte en el curso '3', esta información no se podría conocer hasta que algún alumno se hubiera matriculado en esa asignatura. Debido a que en caso contrario de violaría la integridad de clave al existir valores nulos para la misma.

Borrado de tuplas: se observa que si se eliminan las tuplas correspondientes a los alumnos matriculados en una asignatura, se pierde la información que representa que una asignatura forma parte de un curso.

Actualización de tuplas: si se realiza un cambio de asignación de curso para una asignatura, sería necesario actualizar todas las tuplas correspondientes a todos los alumnos matriculados en esa asignatura. Esto es debido a que existe una gran redundancia de información, ya que en la relación para cada alumno de una asignatura se almacena el curso en que se imparte, y esta información es independiente del alumno que está matriculado en esa asignatura, dependiendo únicamente de la asignatura.

Para eliminar todos estos inconvenientes es necesario realizar un proceso de descomposición de la relación Matricula en dos nuevas tablas que ya cumplen la 2FN. Las tablas o relaciones quedarían así:

MATRICULA (dni, asignatura, apellidos, nombre, nota, aula, lugar)

Imparte (asignatura, curso)

Se puede observar:

- La relación Imparte, se encuentra en la 2FN. La clave de la relación es asignatura, y el atributo curso, depende funcionalmente de forma completa de la clave de la relación.

- Al eliminar el atributo curso en Matricula, se ha eliminado la dependencia funcional no completa entre el atributo curso y la clave de la relación asignatura, y por lo tanto, los problemas que causaba esta dependencia en los procesos de mantenimiento de la información para esa relación.
- No se ha producido pérdida de información, puesto que el atributo asignatura debe definirse como clave foránea en la relación Matricula, es decir, que el atributo asignatura en cualquier tupla de la tabla Matricula sólo podrá tomar valores existentes en alguna tupla de la relación Imparte. Basándonos en esta referencia se podrá conocer en cada momento en qué curso es impartida cada asignatura para cada uno de los cursos.

El mismo razonamiento debe hacerse para la dependencia funcional no completa existente entre la clave de Matricula y apellidos y nombre, ya que estos atributos dependen únicamente del dni y no de la clave completa.

Dni, asignatura → nota, aula, lugar

Dni → Nombre, apellidos

Asignatura → Curso

El nuevo esquema relacional quedaría:

IMPARTE (Asignatura, Curso)

MATRICULA(Dni, Asignatura, Nota, Aula, Lugar)

ALUMNO(Dni, Nombre, Apellidos)

Tercera Forma Normal 1FN

Una relación R cumple la 3FN si y sólo si, cumple la 2FN y cada atributo que no forma parte de la clave de la relación no depende funcionalmente de forma transitiva de la clave primaria de esa relación. Es decir, no pueden existir dependencias entre los atributos que no forman parte de la clave primaria de la relación.

Si observamos la relación MATRICULA, que se encuentra en 2FN, sigue teniendo problemas en los procesos de manipulación de la misma. Los problemas son debidos a que existe una dependencia entre los atributos aula y lugar.

Ejemplo: MATRICULA(Dni, Asignatura, Nota, Aula, Lugar), dependencias:

Dni, Asignatura → Nota, Aula, Lugar

Aula → Lugar

Es decir, que Lugar además de tener una dependencia con la clave completa, tiene una dependencia de aula, por lo que no está en 3FN, originando problemas de manipulación de la relación o tabla. Los problemas son:

Inserción de tuplas: se observa que no se pueden conocer las aulas de cada uno de los lugares utilizados para la docencia de asignaturas hasta que no haya algún alumno matriculado en esa asignatura.

Borrado de tuplas: una vez borrados todos los alumnos matriculados en una asignatura se pierde la información correspondiente a las aulas de cada lugar utilizadas.

Modificación de tuplas: existe una gran redundancia de información, ya que se almacena el lugar en el que se encuentra el aula para cada uno de los alumnos que cursan una asignatura que se imparte en dicha aula.

Estos problemas se originan, debido a la existencia de una dependencia funcional no completa, sino a la existencia de una dependencia funcional transitiva.

Para eliminar los problemas que origina ésta dependencia funcional, la relación se debe descomponer en dos relaciones, quedando el esquema relacional.

IMPARTE (Asignatura, Curso)

MATRICULA(Dni, Asignatura, Nota, Aula)

ALUMNO(Dni, Nombre, Apellidos)

UBICACIÓN (Aula, Lugar)

La relación Matricula, se encuentra en 3FN, ya que todas las dependencias funcionales existentes son completas y todos los atributos no claves dependen de la clave primaria.

La nueva relación generada Ubicación, tiene como clave primaria Aula, con un único atributo Lugar, que depende funcionalmente de la clave, por lo que está en 3FN. En la relación MATRICULA, el atributo Aula debe ser clave ajena o foránea de UBICACIÓN, de forma que este atributo sólo pueda tomar valores nulos o existentes en alguna tupla de la relación UBICACIÓN.

Forma Normal de Boyce-Codd (FNBC)

Las relaciones que cumplen FNBC, también cumplen la 1FN, la 2FN y la 3FN. La FNBC se basa en el concepto de Determinante funcional.

*Se denomina **Determinante funcional** a uno o un conjunto de atributos de una relación R del cual depende funcionalmente de forma completa algún otro atributo de la misma relación.*

*Una relación R cumple la **FNBC** si, y sólo si, se encuentra en 1FN, y cada determinante funcional es una clave candidata de la relación R.*

En el esquema:

IMPARTE (Asignatura, Curso)

MATRICULA(Dni, Asignatura, Nota, Aula)

ALUMNO(Dni, Nombre, Apellidos)

UBICACIÓN (Aula, Lugar)

Todas las relaciones se encuentran en la FNBC, ya que:

- Los únicos determinantes funcionales son las claves para cada una de las relaciones, puesto que en ninguna relación existen claves alternativas.
- En todas las relaciones, las únicas dependencias funcionales son las existentes entre los atributos descriptores de la relación y la clave primaria.

Ejemplo: suponemos el siguiente esquema:

MATRICULA (Dni, Asignatura, Apellidos, Nombre, Nota Curso, Aula, Lugar), en el que consideramos Apellidos, Nombre, Asignatura como una clave candidata de la relación, quedando el esquema

MATRICULA (Dni, Asignatura, Apellidos, Nombre, Nota Curso, Aula, Lugar)

En el esquema existen dos determinantes funcionales, cada uno de ellos compuesto y formado por los agregados: **Dni, Asignatura** y **Apellidos, Nombre, Asignatura**, por lo que tenemos que suponer que no existen dos alumnos con el mismo nombre matriculados en la misma asignatura.

Las dependencias existentes en la relación MATRICULA son:

Asignatura → Curso

(Dni, Asignatura) → Aula

(Dni, Asignatura) → Lugar

(Dni, Asignatura) → Nota

(Apellidos, Nombre, Asignatura) → Aula

(Apellidos, Nombre, Asignatura) → Lugar

(Apellidos, Nombre, Asignatura) → Nota

Aula → Lugar

(Dni, Asignatura) → (Apellidos, Nombre, Matricula)

Se puede observar que existen dependencias entre atributos que no son determinantes funcionales y que es necesario eliminar. La relación debe ser descompuesta, quedando el esquema.

IMPARTE (Asignatura, Curso)

Ubicación (Aula, Lugar)

MATRICULA (Dni, Asignatura, Apellidos, Nombre, Nota, Aula)

Se puede observar que todas las relaciones del esquema se encuentran en 3FN, simplemente por la consideración de los determinantes funcionales y las dependencias que otros atributos de la relación mantienen con ellos. Se han eliminado directamente las dependencias funcionales no completas y las dependencias transitivas.

Las relaciones IMPARTE y UBICACIÓN se encuentran además en FNBC, pues sólo existe un determinante funcional y un atributo dependiente del mismo de forma completa, pero la relación MATRICULA, aunque se encuentra en 3FN, no está en FNBC, existe una dependencia funcional que no se ha tenido en cuenta, es la dependencia funcional mutua existente entre el atributo Dni y el agregado Apellidos + Nombre.

Dni ↔ (Apellidos, Nombre)

En las dos claves candidatas está presente un atributo común, Asignatura, que forma parte de los dos determinantes funcionales, por lo que ambas claves están **traslapadas**.

Dos claves candidatas están **traslapadas** si cada una de ellas está formada por dos o más atributos y alguno de ellos en común en ambas. La existencia de claves traslapadas puede ocasionar problemas en el mantenimiento de la información que hay que resolver.

La relación MATRICULA, sería conveniente descomponerla en dos relaciones que cumplan la FNBC, el esquema quedaría:

IMPARTE (Asignatura, Curso)

Ubicación (Aula, Lugar)

ALUMNO (Dni, Apellidos, Nombre)

MATRICULA (Dni, Asignatura, Nota, Aula)

El esquema obtenido, es similar al obtenido al aplicar las formas normales 1FN, 2FN y 3FN. Con la diferencia que ahora el agregado (Apellidos + Nombre) debe indicarse como clave alternativa, para señalar la dependencia funcional mutua entre Dni y el agregado (Apellidos + Nombre).

OTROS TIPO DE DEPENDENCIAS

Dependencias multivaluadas

Se considera que en una relación R existen dependencias multivaluadas cuando contiene más de un atributo multivaluado.

Definición: en una relación con los atributos X, Y y Z existe una dependencia multivaluada de Y con respecto a X si los posibles valores de Y para un par de valores de X y Z dependen únicamente del valor de X. Se representa: $X \twoheadrightarrow Y$

Ejemplo: Supongamos la relación **Agenda** con la siguiente estructura:

Agenda (nombre, fecha_nacimiento, estado_civil, teléfono, correo)

Cada persona puede tener varios teléfonos, (atributo multivaluado), lo mismo ocurre con el atributo correo, ya que cada persona puede tener más de una dirección de correo, y este atributo es independiente del número de teléfono:

Ahora surgen los problemas, supongamos que nuestra amiga "María", además de los tres números de teléfono, dispone de dos direcciones de correo. ¿Cómo almacenaremos la información relativa a estos datos? Tenemos muchas opciones:

Agenda				
Nombre	FechaNaci	EstadoCivil	Teléfono	Correo
María	13/02/1960	casado	13321232	Maria@sucasa.com
María	13/02/1960	casado	25565445	Maria@sutrabajo.ABC
María	13/02/1960	casado	36635363	Maria@sucasa.com

Si optamos por crear tres tuplas, ya que hay tres teléfonos, y emparejar cada dirección con un teléfono, en la tercera tupla estaremos obligados a repetir una dirección. Otra opción sería usar un *NULL* en esa tercera tupla.

Agenda				
Nombre	FechaNaci	EstadoCivil	Teléfono	Correo
María	13/02/1960	casado	986133212	Maria@sucasa.com
María	13/02/1960	casado	986255654	Maria@sutrabajo.ABC
María	13/02/1960	casado	636635363	NULL

Pero estas opciones ocultan el hecho de que ambos atributos son multivaluados e independientes entre sí. Podría parecer que existe una relación entre los números y las direcciones de correo.

Intentemos pensar en otras soluciones:

Agenda				
Nombre	FechaNaci	EstadoCivil	Teléfono	Correo
María	13/02/1960	casado	986133212	Maria@sucasa.com
María	13/02/1960	casado	986255654	Maria@sucasa.com
María	13/02/1960	casado	636635363	Maria@sucasa.com
María	13/02/1960	casado	986133212	Maria@sutrabajo.ABC
María	13/02/1960	casado	986255654	Maria@sutrabajo.ABC
María	13/02/1960	casado	636635363	Maria@sutrabajo.ABC

Agenda				
Nombre	FechaNaci	EstadoCivil	Teléfono	Correo
María	13/02/1960	casado	986133212	NULL
María	13/02/1960	casado	986255654	NULL
María	13/02/1960	casado	636635363	NULL
María	13/02/1960	casado	NULL	Maria@sucasa.com
María	13/02/1960	casado	NULL	Maria@sutrabajo.ABC

Ahora está claro que los atributos son independientes, pero el precio es crear más tuplas para guardar la misma información, es decir, mayor redundancia.

Pero no sólo eso. Las operaciones de inserción de nuevos datos, corrección o borrado se complican. Ninguna de esas operaciones se puede hacer modificando sólo una tupla, y cuando eso sucede es posible que se produzcan inconsistencias.

Cuarta forma normal (4FN)

La cuarta forma normal tiene por objetivo eliminar las dependencias multivaluadas.

Definición: Una relación está en 4NF si y sólo si, en cada dependencia multivaluada $X \twoheadrightarrow Y$ no trivial, X es clave candidata.

Una dependencia multivaluada $A \twoheadrightarrow B$ es trivial cuando B es parte de A . Esto sucede cuando A es un conjunto de atributos, y B es un subconjunto de A .

Tomemos por ejemplo la tabla de Agenda, pero dejando sólo los atributos multivaluados:

Agenda(nombre, teléfono, correo)

Lo primero que debemos hacer es buscar las claves y las dependencias. Recordemos que las claves candidatas deben identificar de forma unívoca cada tupla. De modo que estamos obligados a usar los tres atributos para formar la clave candidata.

Pero las dependencias que tenemos son:

- nombre \twoheadrightarrow teléfono
- nombre \twoheadrightarrow correo

Y *nombre* no es clave candidata de esta relación.

Resumiendo, debemos separar esta relación en varias (tantas como atributos multivaluados tenga).

Teléfonos(nombre, teléfono)

Correos(nombre, correo)

Ahora en las dos relaciones se cumple la cuarta forma normal.

Quinto Nivel de F/N.

Existe otro nivel de normalización que se aplica a veces, y en la mayoría de los casos no es necesario para obtener la mejor funcionalidad de nuestra estructura de datos o aplicación. Su principio sugiere:

1. La tabla original debe ser reconstruida desde las tablas resultantes en las cuales ha sido troceada.

Los beneficios de aplicar esta regla aseguran que no se ha creado ninguna columna extraña en las tablas y que la estructura de las tablas creadas sea del tamaño justo que tiene que ser. Es una buena práctica aplicar esta regla, pero a no ser que se esté diseñando una estructura de datos muy compleja no será necesaria.

Esta regla puede ser definida de la siguiente forma:

Una relación R satisface la 5FN, también llamada forma normal de proyección (FNPR) sí, y sólo si, toda dependencia de reunión en R está implicada por las claves candidatas entre sí y no por cualquier otro atributo de R, forme o no parte de las claves candidatas.

Ejemplo: si tenemos la relación DOCENCIA (dni, asignatura, aula), y establecemos las siguientes restricciones:

- Un alumno puede estar matriculado en un conjunto de asignaturas.
- Una asignatura se imparte en un conjunto de aulas.
- Cada asignatura es impartida en todas las aulas para un alumno.

Es decir, cada asignatura se puede impartir en varias aulas, y cuando un alumno se matricula de esa asignatura va a recibir clases de esa asignatura en todas las aulas habilitadas para misma, además el alumno se puede matricular de varias asignaturas.

DOCENCIA		
Dni	Asignatura	Aula
15356478	BBDD	1
15356478	BBDD	2
15356478	BBDD	3

En este caso aunque la relación Docencia se encuentra en FNBC y además en la 4FN, ya que ahora se ha considerado que las aulas no son independientes de las asignaturas en las que se matriculan los alumnos, podemos observar que si existe dependencia de reunión entre los atributos de asignatura y aula, esta dependencia da lugar a que exista una gran redundancia en esta relación.

Esta relación puede ser descompuesta sin pérdida de información en tres relaciones dando lugar al siguiente esquema:

DOCENCIA1 (dni, asignatura)

DOCENCIA2 (asignatura, aula)

DOCENCIA3 (dni, aula)

Las tres relaciones se encuentran en la 5FN sin que exista ninguna dependencia de reunión trivial. Además, se puede comprobar que si se realiza la reunión de las relaciones DOCENCIA1, DOCENCIA2, y DOCENCIA3, se obtiene de nuevo la relación DOCENCIA