

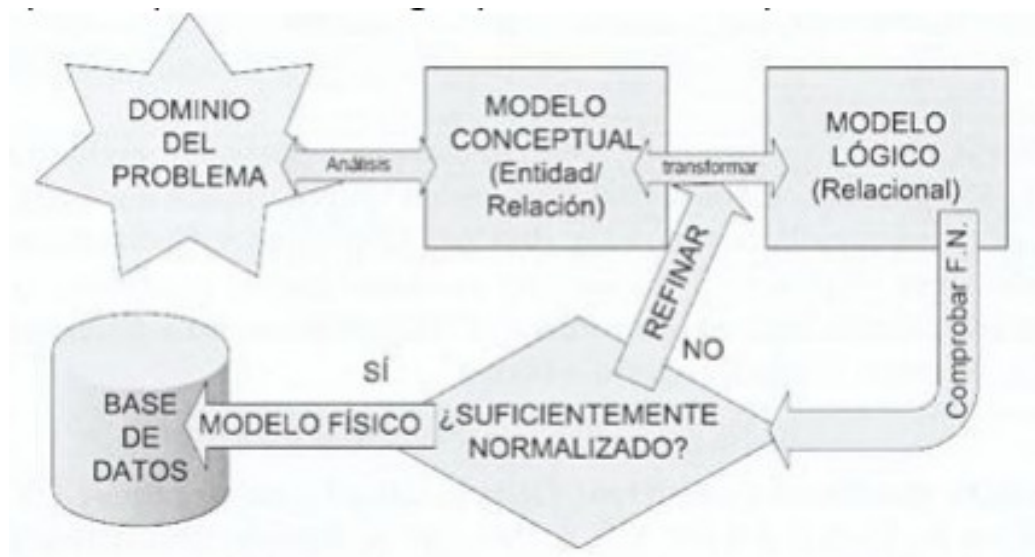
BASES DE DATOS

TEMA 4

NORMALIZACIÓN

1. Introducción

La aplicación de la teoría de la normalización hay que entenderla como una fase de refinamiento que se aplica a un esquema relacional para comprobar que no existe ningún problema de manipulación de los datos



2. Introducción

El proceso de normalización consiste en imponer a las tablas del modelo Relacional una serie de restricciones a través de un conjunto de transformaciones consecutivas.

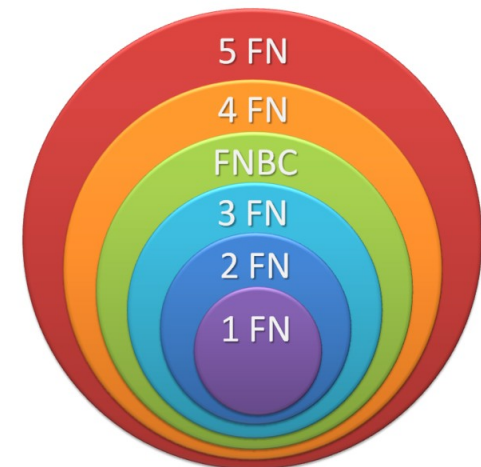
Con el proceso de normalización se alcanzan los siguientes objetivos

1. Almacenar en la base de datos cada hecho solo una vez, es decir, **evitar la redundancia** de datos -> *reduce el espacio de almacenamiento*
2. Que los hechos distintos se almacenen en sitios distintos, **eliminar dependencias** erróneas entre atributos -> *evita anomalías a la hora de operar con los datos*

3. Introducción

El proceso de normalización se basa en el análisis de las dependencias entre atributos. Para ello tendrá en cuenta los conceptos de: **dependencia funcional**, **dependencia funcional completa** y **dependencia transitiva**.

El proceso de normalización se realiza en varias etapas secuenciales. Cada etapa está asociada a **una forma normal**, que establece unos requisitos a cumplir por la tabla sobre la que se aplica. Existen varias formas normales: **Primera, Segunda, Tercera, Boyce-Codd, Quinta y Dominio-Clave**.



4. Introducción

El paso de una forma normal a otra es consecutivo, **si no se satisface una determinada forma normal no puede pasarse al análisis de la siguiente.**

Según vamos avanzando en la normalización, los requisitos a cumplir serán cada vez más restrictivos, lo que hará que nuestro esquema relacional sea cada vez más robusto.

Como norma general, para garantizar que no existan problemas en la actualización de datos, es recomendable aplicar el proceso de normalización hasta **Tercera Forma Normal** o incluso hasta **Forma Normal de Boyce-Codd**.



5. Dependencia funcional

Dependencia es un conjunto de restricciones que se imponen a determinados atributos de las tablas.

Definición

Una **dependencia funcional** entre dos conjuntos de atributos especifica una restricción sobre las posibles tuplas que podrían aparecer en una relación tomando valores estos atributos.

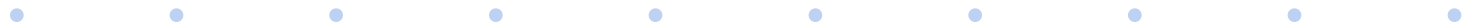
Notación

$R(A) = R(C1:dom1, C2:dom2, C3:dom3 \dots Cn:domN)$

Dada la relación R y el conjunto de atributos C1 y C2 de tipo descriptores,

C2 **depende funcionalmente** de C1 ó C1 **determina a** C2, si a cada valor de C1 le corresponde un valor de C2

C1 → C2



6. Dependencia funcional

Ejemplo:

El escritor "DNI", que se llama "nombre" escribe, solo o en colaboración, el libro de código "ISBN" de título "TITULO" y ha recibido por sus trabajos "EUROS"

HA_ESCRITO(dni:dom_dni,nombre:dom_nom, isbn:dom_isbn, titulo:dom_titulo, euros:dom_euros)

dni → nombre

isbn → título

dni + isbn → euros

• • • • • • • •

7. Tipos de dependencias funcionales

Dependencia funcional completa:

C2 tiene dependencia funcional completa de C1, si depende de C1 pero no de sus subconjuntos.

dni_alumno + cod_asig → nota

dni_alumno + nota

Cod_asig + nota

8. Tipos de dependencias funcionales

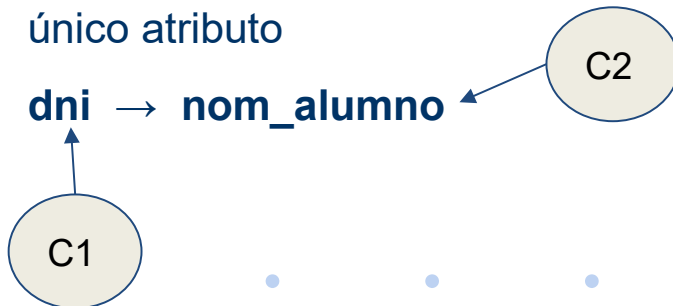
Dependencia funcional trivial:

C2 tiene **dependencia trivial** si $C2 \subset C1$



Dependencia funcional elemental:

C2 tiene **dependencia funcional elemental** de C1 si no es trivial y C2 posee un único atributo



9. Tipos de dependencias funcionales

Dependencia funcional transitiva:

. Si C2 depende funcionalmente de C1 y C3 de C2, pero C1 no depende funcionalmente de C2, se dice entonces que **C3 depende transitivamente de C1**.

Simbólicamente sería:

$C1 \rightarrow C2$ $C2 \rightarrow C3$ entonces $C1 \rightarrow C3$



Fecha de nacimiento → edad

edad → conducir

Fecha de nacimiento → conducir

10. Tipos de dependencias funcionales

Ejercicio:

Dadas las siguientes tablas:

EMPLEADO(DNI, nombre, dirección, localidad, cod_localidad,
nombre_hijo, edad_hijo)

LIBRO(título_libro, num_ejemplar, autor, editorial, precio)

a) Indica qué atributos presentan una dependencia funcional de la clave primaria de la tabla EMPLEADO.

b) Indica qué atributos presentan una dependencia funcional completa en la tabla LIBRO

c) Indica qué atributos presentan una dependencia transitiva en la tabla EMPLEADO

11. Formas normales

El objetivo de la normalización es conseguir **la independencia de los datos**, obteniendo tablas con una estructura óptima.

A partir del conjunto de dependencias funcionales de una relación se deben eliminar los problemas de redundancia.

Optimización de claves

Consiste en minimizar las claves.

La clave principal debe cumplir el criterio de **minimalidad**, es decir, una clave no es mínima si parte de la clave determina al resto de la clave.

Ejemplo:

```
EMPLEADO(código_empleado,dni, nombre,dirección, localidad)
```

11. Formas normales

Conjunto de interés de una relación es un conjunto de dependencias funcionales que basta considerar para normalizar esa tabla eliminando los problemas de redundancia. A partir de este conjunto, se debe poder obtener todas las dependencias funcionales existentes entre los atributos de una tabla. Este conjunto, cumple las siguientes condiciones:

Si $x \rightarrow y$ entonces:
y consta de un solo elemento
y depende completamente de x
el conjunto es minimal

$A = (x, y, a, b)$ Si se decide que la clave es $\{x, y\}$, si x determina a y, entonces la clave no será $\{x, y\}$ sino que únicamente será $\{x\}$.



12. Formas normales

PRIMERA FORMA NORMAL (1FN)

Una tabla está en 1FN si el valor que contiene cada atributo es **único y elemental** (No puede haber atributos multivaluados, es decir, en cada atributo se incluye un elemento).

NO CUMPLE FN1		
Película	Año	Actor
La amenaza Fantasma	1999	Ewan McGregor
		Liam Neeson
		Natalie Portman
Blade Runner	1982	Harrison Ford
		Sean Young
		Rutger Hauer
Avatar	2009	Sam Worthington
		Zoe Saldana
		Sigourney Weaver

CUMPLE FN1		
Película	Año	Actor
La amenaza Fantasma	1999	Ewan McGregor
La amenaza Fantasma	1999	Liam Neeson
La amenaza Fantasma	1999	Natalie Portman
Blade Runner	1982	Harrison Ford
Blade Runner	1982	Sean Young
Blade Runner	1982	Rutger Hauer
Avatar	2009	Sam Worthington
Avatar	2009	Zoe Saldana
Avatar	2009	Sigourney Weaver

13. Formas normales

PRIMERA FORMA NORMAL (1FN)

Ejemplo:

Proveedor = vcod + nombre + {teléfonos} + dirección
CP{vcod}

vcod	nombre	Teléfonos	dirección
v1	Pepe	(962452555, 912588585)	C/Calle principal, nº2, Madrid
v2	juan	(936969669, 656655445, 914547441)	C/Mayor, nº3), Barcelona

No está en 1FN

• • • • • • • • • •

14. Formas normales

PASO A 1FN

1. Se crea, a partir de la tabla inicial, una nueva tabla cuyos atributos son los que presentan dependencia funcional de la clave primaria. La clave de esta tabla será la misma clave primaria de la tabla inicial. Esta tabla ya estará en 1FN.
2. Con los atributos restantes se crea otra tabla y se elige entre ellos uno que será la clave primaria de dicha tabla.
3. Comprobaremos si esta segunda tabla está en 1FN. Si es así, la tabla inicial ya estará normalizada a 1FN y el proceso termina. Si no está en 1FN, tomaremos la segunda tabla como tabla inicial y repetiremos el proceso.

Paso a 1FN Ejemplo:

Proveedor = vcod + nombre + calle + n° + población CP(vcod)

Telefono = vcod + telefono CP{vcod + telefono}

15. Formas normales

SEGUNDA FORMA NORMAL (2FN)

Una tabla está en 2FN si está en 1ª Forma Normal y todo atributo que no pertenece a la clave primaria tiene **dependencia funcional completa** de toda la clave y no de parte de ella.

Únicamente cabe la posibilidad de que una tabla no está en 2FN si su clave primaria está formada por más de un atributo.

Ejemplo:

COMPRAS (CódigoProducto, CódigoProveedor, NombreProducto, Cantidad, FechaCompra)

Dependencia funcional **CódigoProducto -> NombreProducto**, por tanto, al no ser dependencia funcional completa, no está en FN2

16. Formas normales

PASO A 2FN

Cuando la clave está formada por más de un campo se necesitará crear nuevas tablas para eliminar las dependencias funcionales:

1. Se crea, a partir de la tabla inicial, una nueva tabla con los atributos que dependen funcionalmente de forma completa de la clave. La clave de esta tabla será la misma clave primaria de la tabla inicial. *Esta tabla ya estará en 2FN.*
2. Con los atributos restantes, se crea otra tabla que tendrá por clave el subconjunto de atributos de la clave inicial de los que dependen de forma completa. Se comprueba si esta tabla está en 2FN. Si es así, la tabla inicial ya está normalizada y el proceso termina. Si no está en 2FN, tomamos esta segunda tabla como tabla inicial y repetiremos el proceso.

17. Formas normales

PASO A 2FN

Ejemplo:

COMPRAS (CódigoProducto, CódigoProveedor, NombreProducto,
Cantidad, FechaCompra)



COMPRAS (CódigoProducto, CódigoProveedor, Cantidad, FechaCompra)

PRODUCTO (CódigoProducto, NombreProducto)

• • • •

18. Formas normales

PASO A 2FN

Ejemplo2:

Ha_escrito = dni , nombre, isbn, titulo, euros

{dni, isbn }

(Clave principal)

dni \rightarrow nombre

dni + isbn \rightarrow euros

isbn \rightarrow titulo

} (Dependencias funcionales)

Como consecuencia de esa redundancia, la manipulación de esta relación presenta los siguientes problemas:

- **Inserción:** No puede darse de alta una autor que no haya escrito ningún libro ni tampoco introducir la información de un libro si no se sabe quién lo ha escrito.
- **Borrado:** El borrado de una tupla puede provocar a su vez el borrado de toda la información de un autor si éste sólo ha escrito u libro o de un libro si sólo tiene un autor.
- **Actualización:** La redundancia asociada a los atributos nombre y título provoca que la actualización de los mismos sea mucho más costosa.

19. Formas normales

PASO A 2FN

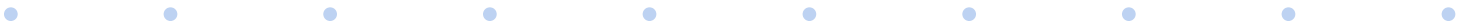
Ejemplo2:

Tablas normalizadas en 2FN:

Ha escrito= dni + isbn + euros

Autor = dni + nombre

Libro = isbn + título



20. Formas normales

TERCERA FORMA NORMAL (3FN)

Una tabla está en 3ª forma normal si está en 2ª Forma Normal y **no existen atributos que no pertenezcan a la clave primaria que puedan ser conocidos mediante otro atributo que no forme parte de la clave primaria**, es decir, si no existen dependencia funcionales transitivas.

El objetivo de pasar a 3FN es el hacer que todo los atributos de una tabla dependan únicamente de la clave principal o alternativa, con ello evitamos introducir redundancia en los datos.



20. Formas normales

Ejemplo:

PRODUCTOS (CódigoProducto, Nombre, Fabricante, País)

Dependencia funcional

CódigoProducto \rightarrow CódigoFabricante

CódigoFabricante \rightarrow País

CódigoProducto \rightarrow País

CódigoFabricante depende transitivamente de CódigoProducto, por tanto, **no está en tercera forma normal.**



21. Formas normales

PASO A 3FN

1. Se crea, a partir de la tabla inicial, una nueva tabla con los atributos que no poseen dependencias transitivas de la clave primaria. Esta tabla ya estará en 3FN.
2. Con los atributos restantes, se crea otra tabla con los dos atributos no clave que intervienen en la dependencia transitiva, y se elige uno de ellos como clave primaria, si cumple los requisitos para ello. Se comprueba si esta tabla está en 3FN. Si es así, la tabla inicial ya está normalizada y el proceso termina. Si no está en 3FN, tomamos esta segunda tabla como tabla inicial y repetiremos el proceso.

Paso a 3FN Ejemplo:

PRODUCTOS (CódigoProducto, Nombre, CodigoFabricante)

FABRICANTE (CodigoFabricante, Pais)

• • • • • • • • • •

22. Formas normales

PASO A 3FN

Ejemplo 2:

Escritor = dni + nombre + dirección + CP + distrito

{dni} (Clave principal)

Dependencias funcionales:

- dni → nombre
- dni → dirección
- dni → CP
- dni → distrito
- CP → distrito

Debido a la redundancia por esta dependencia, la manipulación de esta relación presenta los siguientes problemas.

- **Inserción:** No puede introducirse la información de cuál es el código de cada población hasta que no exista un escritor de esa población.
- **Borrado:** El borrado de un escritor puede implicar implícitamente la pérdida de la información de la población si el escritor es el único en esa población.
- **Actualización:** La modificación del código de una población hay que realizarla tantas veces como escritores vivan en esa población, ya que esa información es redundante y se replica

Tablas normalizadas en 3FN:

Escritor = dni + nombre + dirección + CP • • • • •

Distritos = CP + distrito

23. Formas normales

TERCERA FORMA NORMAL

EXCEPCIÓN

Una tabla sí que está en 3FN si la dependencia funcional es entre un atributo que no es clave y la clave principal.

En este caso estaríamos hablando de una clave alternativa. (SI está en 3 FN)

ALUMNO= Nexpd + Nombre + Edad + dni + dirección

dni \rightarrow Nexpd (dni, es una clave alternativa)



24. Formas normales

FORMA NORMAL BOYCE CODD (FNBC)

Una tabla está en FNBC si está en 3ª Forma Normal si solo existen dependencias funcionales elementales que dependen de la clave primaria o cualquiera alternativa.

Si la clave primaria es simple y está en 3FN estará en FNBC.

Ejemplo:

TUTORIA (DNIAlumno, asignatura, tutor)

DniAlumno+asignatura -> tutor

Restricción: un tutor solo tutoriza un asignatura

tutor-> asignatura

la tabla está en 3FN porque no hay dependencias funcionales transitivas, y sin embargo no está en FNBC porque hay un determinante que no es clave

TUTORÍAS		
<u>DNI</u>	<u>Asignatura</u>	<u>Tutor</u>
12121219A	Lenguaje	Eva
12121219A	Matemáticas	Andrés
3457775G	Lenguaje	Eva
5674378J	Matemáticas	Guillermo
5674378J	Lenguaje	Julia
5634823H	Matemáticas	Guillermo

25. Formas normales

FORMA NORMALBOYCE CODD (FNBC)

Aquellas tablas en las que todos sus atributos forman parte de la clave primaria, están en FNBC. Por lo tanto, si encontramos un determinante que no es clave candidata, la tabla no estará en FNBC.

Esta redundancia suele ocurrir por una mala elección de la clave.

Para normalizar a FNBC tendremos:

a) descomponer la tabla inicial en dos, siendo cuidadosos para evitar la pérdida de información en dicha descomposición.

Paso a FNBC Ejemplo:

TUTORIA (DNIAlumno, tutor)

TUTOR (tutor, asignatura)



26. Formas normales

FORMA NORMALBOYCE CODD (FNBC)

SUMINISTRA (empresa, producto, distribuidor)

- Cada empresa y producto es suministrado por un único distribuidor.
empresa+producto-> distribuidor
- Cada distribuidor suministra un solo producto.

Distribuidor -> producto

- Cada producto es vendido por
- varias empresas

DISTRIBUIDOR (distribuidor, producto)

SUMINISTRA (empresa, distribuidor)

EMPRESA	PRODUCTO	DISTRIBUIDOR
EROSKI	GALLETAS	CUETARA
EROSKI	ZUMO	DON SIMÓN
DIA	GALLETAS	CUETARA

27. Formas normales

ELIMINAR REDUNDANCIAS DE LAS TABLAS OBTENIDAS

REDUNDANCIA ENTRE CAMPOS:

Eliminar uno o varios campos que son redundantes unos con otros. Se eliminará la redundancia en el caso en que efectuar las modificaciones no se produzcan errores.

REDUNDANCIA INTERTABLA:

Cuando los campos de una tabla son deducibles a partir de información almacenada en otras tablas. Tener en cuenta que eliminar la redundancia puede provocar ralentización de las consultas.

FUSIONAR TABLAS.

Si a lo largo del proceso de normalización, se obtiene tablas que comparten la misma clave primaria, estas tablas son susceptibles a ser fusionadas en una solo añadiendo los atributos en una que la otra posee diferentes.