

UNIDAD 3: NORMALIZACIÓN

Índice

1.	Normalización	1
1.1	Formas Normales	3
1.2	Dependencias funcionales	4
1.2.1	Dependencia funcional completa (o total)	5
1.2.2	Dependencia funcional transitiva	5
1.2.3	Atributos primos y no primos	5
1.2.4	Primera forma normal (1FN)	6
1.2.5	Segunda forma normal (2FN)	6
1.2.6	Tercera forma normal (3FN)	7
1.2.7	FORMA NORMAL DE BOYCE Y CODD	8

1. Normalización

Una vez obtenido el esquema relacional resultante del esquema entidad/relación que representa la base de datos, normalmente tendremos una buena base de datos. Pero otras veces, debido a **fallos en el diseño** o a problemas indetectables, tendremos un esquema que puede producir una base de datos que incorpore problemas que dificultan tanto el procesamiento y el mantenimiento de la base de datos.

Vamos a ver esto con un ejemplo: Partiendo de la relación ESCRIBE

ESCRIBE (AUTOR, PAIS_AUTOR, COD_LIBRO, TITULO, EDITORIAL, AÑO)

<u>Autor</u>	<u>Pais_autor</u>	<u>Cod_libro</u>	<u>Titulo</u>	<u>Editorial</u>	<u>Año</u>
Date	EEUU	989877	Databases	Addison-Wesley	1990
Date	EEUU	977777	SQL Standard	Addison-Wesley	1986
Date	EEUU	98967	A guide to INGRESS	Addison-Wesley	1988
Codd	EEUU	7890	Relational Mod.	Addison-Wesley	1988
Gardarin	Francia	12345	Bases de Datos	Paraninfo	1986
Gardarin	Francia	67890	Comparación BD	Eyroyes	1984
Valduriez	Francia	67890	Comparación BD	Eyroyes	1984
Kim	EEUU	12333	O-O Databases	ACM Press	1989
Lochovsky	Canadá	12333	O-O Databases	ACM Press	1989

La **relación ESCRIBE** almacena datos sobre autores y sus libros. Presenta los **siguientes problemas**:

- **Redundancias**: **consiste en la repetición innecesaria de datos**.

Ej: para un autor determinado → país del autor

para un libro determinado → editorial y año.

Si no está bien diseñada nuestra base de datos puede ocasionar **problemas al insertar nuevas tuplas, al actualizar o borrar**. La **redundancia de datos** puede ocasionar que nuestra base de datos sea **inconsistente**.

Anomalías.

- **Anomalías de modificación**, debidas a las redundancias. **Repetición de la misma información en tuplas diferentes** y consiguiente en la **necesidad de propagar actualizaciones** en todas las tuplas. **Problemas de inconsistencia** si se cambia en unas y en otras no.

Ej: Se repite la editorial para cada libro. Cambiar la editorial para un autor del libro XXX y no para otro autor de ese mismo libro.

- **Anomalías de inserción**: **Imposibilidad de dar de alta una tupla** por **no disponer del valor de un atributo principal**.

Ej: No se puede introducir un autor sin libros, ni una obra anónima, y la inserción de un libro con varios autores precisa de la inserción de una tupla por autor con los mismos datos de libro en cada una de ellas.

- **Anomalías de borrado**: **Pérdida de información por dar de baja una tupla**.

Ej: Al borrar un libro se eliminan de la BD también sus autores (si no existen en algún otro libro), y viceversa

La solución que inmediatamente surge es la descomposición del esquema de la relación anterior. Sin embargo es necesario **descomponer de forma adecuada** o se pueden producir problemas.

En cualquier caso la mayor parte de problemas se agravan si no se sigue un modelo conceptual y se decide crear directamente el esquema relacional. En ese caso, el diseño tiene una garantía casi asegurada de funcionar mal.

Cuando **aparecen los problemas** enumerados, entonces **se les puede resolver usando reglas de normalización**. Estas reglas suelen forzar la división de una tabla en dos o más tablas para arreglar ese problema.

Se puede, al **diseñar una base de datos**, partir del **esquema conceptual** y aplicar posteriormente la **Teoría de Normalización**, o bien podemos **emplear directamente la teoría de Normalización** partiendo de un esquema de **relación universal** (esto es, un único esquema de relación que posee todos los atributos de la base de datos) y dividirlos en los diferentes esquemas de relación.

Si se **siguiera la metodología de diseño propuesta**, realizando 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.

Para saber si un determinado esquema relacional es o no correcto, será preferible aplicar siempre a dicho esquema un **método formal de análisis** que determine lo que pueda estar equivocado en el mismo y nos permita llegar a otro esquema en el que se asegure el cumplimiento de ciertos requisitos; este método formal, como ya hemos indicado, es **la teoría de la normalización**.

La **teoría de la normalización evita** las **redundancias y las anomalías** obteniendo relaciones más estructuradas que no presenten los problemas que comentábamos anteriormente.

Así, en lugar de la relación del ejemplo que aparecía en la tabla universal anterior, se podría haber diseñado el siguiente esquema relacional:

LIBRO (Cod libro, Titulo, Editorial, Año)

AUTOR (Nombre, Nacionalidad)

ESCRIBE (Cod libro, Nombre)

donde se ha seguido el principio básico anteriormente enunciado, separando hechos distintos en relaciones distintas, de forma que cada uno de estos esquemas de relación recoge un hecho bien determinado y concreto del mundo real con sus correspondientes atributos; esto es, evidentemente, lo que habría hecho cualquier diseñador.

1.1 Formas Normales

La **teoría de normalización** se centra en lo que se conoce con **formas normales**.

Se dice que un esquema de relación está en una **determinada forma normal** si satisface un **conjunto específico de restricciones**.

- En el proceso de normalización, según la propuesta original de **CODD (1977)** se somete **al esquema de relación a una serie de pruebas para “certificar” si pertenece o no a una cierta forma normal**.
- En un principio, **Codd propuso tres formas normales**, a las cuales llamó **1FN** (Primera Forma Normal), **2FN** (Segunda Forma Normal) y **3FN** (Tercera Forma Normal). Posteriormente, CODD y BOYCE propusieron una definición más estricta de 3FN, a la que se conoce como el nombre de forma Normal de **Boyce-Codd**.
- Todas estas formas normales se basan en el concepto de **dependencias funcionales**.

Más adelante se propusieron otras formas normales **4FN y la 5FN**, con el fundamento de las **dependencias multivaluadas**.

Una tabla puede encontrarse en primera forma normal y no en segunda forma normal, pero no al contrario. Es decir los **números altos de formas normales son más restrictivos** (la quinta forma normal cumple todas las anteriores).

NF	¿Qué significa?
1NF	No hay grupos repetidos
2NF	No hay dependencias parciales
3NF	No hay dependencias transitivas
BCNF	No hay DFs con determinantes no-claves



Hay que tener en cuenta que muchos diseñadores opinan que basta con llegar a la forma Boyce-Codd, ya que la cuarta, y sobre todo la quinta, forma normal es polémica. Hay quien opina que hay bases de datos peores en quinta forma normal que en tercera. En cualquier caso debería ser obligatorio para cualquier diseñador llegar hasta la forma normal de Boyce-Codd.

1.2 Dependencias funcionales

El diseño de esquemas se basa en el concepto de **dependencia de datos**, es decir, **la capacidad de determinar el valor de un atributo conociendo otro(s)**.

Por ejemplo, la dirección de un alumno depende de su nº de expediente, es decir, el atributo Dirección depende del atributo nº de expediente.

Se dice que un **conjunto de atributos (Y) depende funcionalmente de otro conjunto de atributos (X)** si para **cada valor de X hay un único valor posible para Y**. Simbólicamente se denota por **$X \rightarrow Y$** .

Al **conjunto X** del que depende funcionalmente el conjunto Y se le llama **determinante**. Al **conjunto Y** se le llama **implicado**.

Por ejemplo, en la relación ALUMNOS:

ALUMNOS

NUM-ALUMNO	NOMBRE	APELLIDOS	EDAD
0001	Carmen	García Casas	16
0002	Ana	Sánchez León	17
0003	María	Vázquez Rivas	16
0004	Ana	Pérez Fuertes	18
0005	Luis	López Calle	16
0006	Carlos	López Calle	17

Vemos que NUM-ALUMNO \rightarrow NOMBRE, NUM-ALUMNO \rightarrow APELLIDOS, NUM-ALUMNO \rightarrow EDAD. Pero NUM-ALUMNO no depende funcionalmente de EDAD (EDAD \rightarrow NUM-ALUMNO) ya que puede haber varios alumnos que tengan esa edad.

TAREA_1: Estudia si existen las siguientes dependencias funcionales en la tabla de actividades:

ACTIVIDADES

NUM-ALUMNO	COD-ACTIVIDAD	NOM-ACTIVIDAD	CUOTA
0001	2	NATACION	2000
0002	2	NATACION	2000
0002	3	PATINAJE	2000
0003	1	TENIS	4000
0003	2	NATACION	2000
0004	1	TENIS	4000
0005	1	TENIS	4000
0005	2	NATACION	2000

NUM-ALUMNO \rightarrow COD-ACTIVIDAD

COD-ACTIVIDAD \rightarrow NUM-ALUMNO

NUM-ALUMNO \rightarrow CUOTA

COD-ACTIVIDAD \rightarrow NOM-ACTIVIDAD

COD-ACTIVIDAD \rightarrow CUOTA

TAREA_2: Dadas las siguientes relaciones estudia las dependencias funcionales

LIBRO (ISBN, Título, Autor, Editorial, Fecha-edición)

SOCIO (Código, Nombre, Dirección, Teléfono)

PRESTAMO (ISBN, Código, Fecha-préstamo, Fecha-devolución)

Para **determinar las dependencias funcionales** de una base de datos es necesario conocer el **significado de sus atributos, su función y las reglas de funcionamiento de la base**.

Si en el ejemplo anterior existiera una regla que estableciese que un autor sólo trabaja para una editorial, aparecería una nueva dependencia: Autor \rightarrow Editorial.

1.2.1 Dependencia funcional completa (o total)

Se dice que Y tiene una **dependencia funcional total** de X si y sólo si no existe un subconjunto tal que $Z \rightarrow Y$, es decir, si Y depende funcionalmente de X y no dependen funcionalmente de ningún subconjunto de propio de X (si de esa *colección de atributos que está formado X no podemos eliminar ninguno de ellos sin que se rompa la dependencia funcional*)

- Por ejemplo, en $F = \{A \rightarrow B\}$, B tiene una dependencia total de A y, sin embargo, en $F = \{AC \rightarrow B \text{ y } A \rightarrow B\}$, B no tiene dependencia total de AC, porque la tiene también de A.
- Por ejemplo en una tabla de clientes, el conjunto de atributos formado por el nombre y el dni producen una dependencia funcional sobre el atributo apellidos. Pero no es plena ya que el dni individualmente, también produce una dependencia funcional sobre apellidos. El dni sí produce una dependencia funcional completa sobre el campo apellidos.

1.2.2 Dependencia funcional transitiva

Se produce cuando tenemos tres conjuntos de atributos X, Y y Z. **Y depende funcionalmente de X** ($X \rightarrow Y$), **Z depende funcionalmente de Y** ($Y \rightarrow Z$). Además X no depende funcionalmente de Y. Entonces ocurre que **X produce una dependencia funcional transitiva sobre Z**.

Si $X \rightarrow Y$ e $Y \rightarrow Z$, entonces, $X \rightarrow Z$.

Ej: ISBN \rightarrow TITULO

TITULO \rightarrow AUTOR entonces ISBN \rightarrow AUTOR

Ej: si X es el atributo Código de libro, e Y es el atributo Editorial. Entonces $X \rightarrow Y$ (para cada libro, sabemos la editorial que lo publica). Si Z representa el pais de la editorial (suponiendo que una editorial tiene su sede en un único pais) , entonces $Y \rightarrow Z$ (el pais depende funcionalmente de la editorial). Como ocurre que $Y \not\rightarrow X$ (la editorial no me permite conocer un libro en concreto). Entonces $X \rightarrow Z$ (el pais depende transitivamente del código del libro).

1.2.3 Atributos primos y no primos

- Un atributo de relación se denomina **atributo primo** de una relacion si es miembro de **cualquier clave candidata** de la relación.
- Un atributo es **no primo** si **no es miembro de ninguna clave candidata**.

Ej: **TRABAJA-EN** (NSS,NUM-PROYECTO, HORASEMALES)

Clave (NSS,NUM-PROYECTO)

Atributos primos : NSS y NUM-PROYECTO

Atributo no primo: HORASEMANALES

1.2.4 Primera forma normal (1FN)

Se dice que una tabla se encuentra en primera forma normal si impide que un atributo de una tupla pueda tomar más de un valor (**todos sus atributos son atómicos**). No se aceptan atributos multivalorados.

La tabla:

Cod_libro	Titulo	Autor
989877	Databases	Date
67890	Comparación BD	Gardarin Valduriez
12333	O-O Databases	Kim Lochovsky

Visualmente es una tabla, pero no una *tabla relacional* (lo que en terminología de bases de datos relacionales se llama **relación**). **No cumple la primera forma normal**.

Para **estar en primera forma normal** debería ser:

Cod_libro	Titulo	Autor
989877	Databases	Date
67890	Comparación BD	Gardarin
67890	Comparación BD	Valduriez
12333	O-O Databases	Kim
12333	O-O Databases	Lochovsky

1.2.5 Segunda forma normal (2FN)

Un esquema de relación R está **en 2ª Forma normal** cuando está en **1FN** y además **todo atributo no primo A en R dependen funcionalmente de manera completa** con respecto de cada una de las **claves candidatas** de R

Deben depender funcionalmente de la totalidad de cada una de las claves candidatas, es decir, **no se permite que un atributo dependa sólo de un subconjunto de atributos de una de las claves**.

Si una relación no se encuentra en 2FN se puede aplicar el siguiente teorema de manera que, al descomponer la relación original en dos, por lo menos una de ellas se encuentre en 2FN.

Teorema I.

Sea una relación **R(A, B, C, D)** con clave **(A, B)** y tal que **R.A → R.D**.

Entonces la relación R **puede descomponerse** como:

$$R \begin{cases} R1 (A, D) \\ R2 (A, B, C) \end{cases}$$

Ejemplo:

ALUMNOS

Num_exp	Num_Materia	Nombre	Apellido1	Apellido2	Nota
0123	34	Pedro	Valiente	López	9
0123	25	Pedro	Valiente	López	8
0135	34	Ana	Fernández	Paz	9
0201	25	Sara	Crespo	Leis	5
0201	34	Sara	Crespo	Leis	8

Suponiendo que **el nº de expediente y el número de materia** formen una **clave principal para esta tabla**, sólo la nota tiene dependencia funcional completa. El nombre y los apellidos dependen de forma completa del nº de expediente. La tabla no es 2FN, para arreglarlo:

ALUMNOS

Num_exp	Nombre	Apellido1	Apellido2
0123	Pedro	Valiente	López
0135	Ana	Fernández	Paz
0201	Sara	Crespo	Leis

NOTAS

Num_exp	Num_Materia	Nota
0123	34	9
0123	25	8
0135	34	9
0201	25	5
0201	34	8

1.2.6 Tercera forma normal (3FN)

Una relación está en **3ª Forma Normal** si está en **2FN** y **ningún atributo no primo de R depende transitivamente de alguna clave**. Es decir, no pueden existir dependencias transitivas entre los atributos que no forman parte de alguna clave de la relación R.

Este enunciado significa que la dependencia de Y con respecto a la clave X no se hace a través de un tercer atributo Z sino directamente:

Si una relación no se encuentra en 3FN se puede aplicar el siguiente teorema:

Teorema II

Sea una relación **R(A,B,C)** con **clave (A)** y tal que **R.B → R.C**.

Entonces la relación R puede **descomponerse** como:

$$R \Rightarrow \left\{ \begin{array}{l} R1(\underline{A}, B) \\ R2(\underline{B}, C) \end{array} \right.$$

ALUMNOS

Num_exp	Nombre	Apellido1	Apellido2	Cod_Provincia	Provincia
0123	Pedro	Valiente	López	08	Barcelona
0124	Paulina	Vidal	López	27	Lugo
0135	Ana	Fernández	Paz	36	Pontevedra
0198	Serafín	Caamaño	Loureiro	27	Lugo
0201	Sara	Crespo	Leis	36	Pontevedra

La provincia depende funcionalmente del código de provincia (**Cod_Provincia** → **Provincia**) lo que hace que **no esté en 3FN**. El arreglo sería:

ALUMNOS

Num_exp	Nombre	Apellido1	Apellido2	Cod_Provincia
0123	Pedro	Valiente	López	08
0124	Paulina	Vidal	López	27
0135	Ana	Fernández	Paz	36
0198	Serafín	Caamaño	Loureiro	27
0201	Sara	Crespo	Leis	36

PROVINCIA

Cod_Provincia	Provincia
08	Barcelona
27	Lugo
36	Pontevedra

1.2.7 FORMA NORMAL DE BOYCE Y CODD

Las tres formas normales anteriores fueron expuestas originalmente por CODD(1970), con el paso del tiempo se mostraron insuficientes para afrontar ciertos problemas en relaciones que presentaban varias claves candidatas compuestas por varios atributos donde estos atributos se solapaban. Por ello, Boyce y Codd definieron la llama forma normal que lleva su nombre(FNBC) y se trata de una forma más estricta de la 3FN.

Las relaciones que satisfacen la restricción impuesta por la forma normal FNBC satisfacen la FN2 y FN3.

Una relación está en **FNBC** si y sólo si está en 3FN y todos **determinantes funcionales son claves candidatas**.

La 2FN y la 3FN eliminan las dependencias parciales y las dependencias transitivas de las claves. Pero este tipo de dependencias todavía pueden existir sobre otras claves candidatas, si las hubiera. La FNBC es más fuerte que la 3FN, por lo tanto, toda relación en FNBC está en 3FN.

La violación de la FNBC es poco frecuente ya que se da bajo ciertas condiciones que raramente se presentan. Se debe comprobar si una relación viola la FNBC si tiene **dos o más claves candidatas compuestas** que tienen al menos un atributo en común.

En realidad son muy pocos los casos de relaciones que se encuentran en 3FN y no están en FNBC. Este tipo de tablas son aquellas en las que se dan las siguientes circunstancias:

- Existan varias claves candidatas
- Las claves candidatas son compuestas
- Las claves candidatas se solapan (tienen por lo menos un atributo común)

En estos casos se puede aplicar el siguiente teorema, de manera que al descomponer la relación original en dos por lo menos una de ellas se encuentre en FNBC.

Teorema III.

Sea una relación $R(\underline{A}, \underline{B}, C, D)$ con claves candidatas (A, B) y (B, C) y tal que

$A \rightarrow B$. Entonces la relación R puede descomponerse de cualquiera de las dos siguientes maneras:

$$R \rightarrow \begin{cases} R1(\underline{A}, C) \\ R2(\underline{B}, C, D) \end{cases}$$

$$R \rightarrow \begin{cases} R1(\underline{A}, C) \\ R2(\underline{A}, B, D) \end{cases}$$

Ejemplo 1:

En nuestro supuesto, **un trabajador puede trabajar en varios departamentos**. En dicho departamento hay **varios responsables**, pero cada **trabajador sólo tiene asignado uno**. El detalle importante que no se ha tenido en cuenta, es que el o la responsable sólo puede ser responsable en un departamento.

Las claves candidatas serían:

CP: {trabajador, departamento} CA: {trabajador, responsable}

ORGANIZACIÓN

Trabajador	Departamento	Responsable
Alex	Producción	Felipa
Arturo	Producción	Martín
Carlos	Ventas	Julio
Carlos	Producción	Felipa
Gabriela	Producción	Higinio
Luisa	Ventas	Eva
Luisa	Producción	Martín
Manuela	Ventas	Julio

Pero el detalle de que **responsable sólo puede ser responsable en un departamento** detalle último produce una dependencia funcional ya que: **Responsable \rightarrow Departamento**

Por lo tanto *hemos encontrado un determinante que no es clave candidata*. Por tanto, no está en FNBC. En este caso la redundancia ocurre por mala selección de clave. La redundancia del departamento es completamente evitable. La solución sería:

➤ RESPONSABLE (Responsable, Departamento)
 PERSONAL (Trabajador, Responsable)

PERSONAL

<u>Trabajador</u>	<u>Responsable</u>
Alex	Felipa
Arturo	Martín
Carlos	Julio
Carlos	Felipa
Gabriela	Higinio
Luisa	Eva
Luisa	Martín
Manuela	Julio

RESPONSABLE

<u>Responsable</u>	Departamento
Felipa	Producción
Martín	Producción
Julio	Ventas
Higinio	Producción
Eva	Ventas