

## CC20A

# Apuntes de Normalización

Cuando se diseña una base de datos mediante el modelo relacional, al igual que ocurre en otros modelos de datos, tenemos distintas alternativas. Es decir, podemos obtener diferentes esquemas relacionales y no todos son equivalentes, ya que algunos van a representar la realidad mejor que otros. Es necesario conocer qué propiedades debe tener un esquema relacional para representar adecuadamente una realidad y cuáles son los problemas que se pueden derivar de un diseño inadecuado.

La teoría de la Normalización es un método objetivo y riguroso que se aplica en el diseño de bases de datos relacionales. La normalización se usa para ver si una tabla está bien o mal diseñada. Una tabla está bien diseñada si no tiene redundancia (datos repetidos) y está mal en caso contrario.

Algunos problemas que se pueden presentar son:

- Incapacidad para almacenar ciertos hechos.
- Redundancias y por tanto, posibilidad de incoherencias.
- Ambigüedades.
- Pérdida de información.
- Pérdida de dependencias funcionales, es decir, ciertas restricciones de integridad que dan lugar a interdependencias entre los datos.
- Aparición en la BD de estados no válidos, es decir, anomalías de inserción, borrado y modificación.

Por ejemplo, analicemos la siguiente tabla con información de los empleados de cierta compañía.

**EMP**

NEmp	NomEmp	NJefe	NomJefe	NDpto	NomDpto	NProj	NomProj	FecIni
7369	Smith	7902	Ford	20	Investigación	15	Factibilidad	10/05/2001
7369	Smith	7902	Ford	20	Investigación	35	Pruebas	20/05/2001
7369	Smith	7902	Ford	20	Investigación	45	Control	20/06/2001
7499	Allen	7698	Blake	30	Ventas	15	Factibilidad	05/05/2001
7499	Allen	7698	Blake	30	Ventas	25	Análisis	15/05/2001
7499	Allen	7698	Blake	30	Ventas	45	Control	20/06/2001

Algunos problemas que se presentan son los siguientes:

- *Redundancia.* El nombre del empleado, número, nombre del jefe, etc., se repiten por cada ocurrencia del mismo empleado. Lo mismo sucede cuando en un proyecto ha trabajado más de un empleado, se repite el número y nombre del proyecto.
- *Anomalías de modificación.* Es fácil cambiar el nombre de un proyecto en una tupla sin modificar el resto de las que corresponden al mismo proyecto, lo que da lugar a incoherencias.

- *Anomalías de inserción.* Si queremos ingresar información de un nuevo proyecto, en el que no hubiera todavía ningún empleado asignado, no sería posible. La inserción de un empleado que trabaja en dos o más proyectos, obliga a insertar dos o más tuplas en la relación.
- *Anomalías de borrado.* Si queremos eliminar un cierto departamento, deberíamos perder los datos de sus empleados y viceversa.

Las formas normales son pautas que ayudan a obtener tablas con menos redundancia y sin anomalías. Una forma normal es una “nota” que se le pone a una tabla de acuerdo al grado de redundancia que presenta.

Las formas normales son 1FN, 2FN, 3FN, 4FN, ordenadas desde la menos exigente a la más exigente. Existen formas normales más exigentes aún, pero en el trabajo práctico no se usan.

Para definir la 2FN y 3FN debemos saber primero el significado de los conceptos de dependencia funcional y llaves primarias de un esquema de relación.

## Dependencia Funcional

En una tabla la columna Y depende funcionalmente de la columna X si cada valor de la columna Y está determinado por el valor de la columna X en la misma fila.

Gráficamente :  $X \rightarrow Y$

Es decir, para todo par de filas en la tabla, tales que tienen igual valor en la columna X, entonces también deben tener igual valor en la columna Y.

Tanto el determinante (a la izquierda de la flecha, X) como el dependiente (a la derecha de la flecha, Y) pueden ser compuestos (más de una columna).

Las dependencias funcionales son una propiedad del mundo real representado mediante la tabla. No son una propiedad del contenido de la tablas en un instante.

Formalmente: Sea el esquema de relación **R** definido sobre el conjunto de atributos **A** y sean **X** e **Y** subconjuntos de **A** llamados *descriptores*. Se dice que **Y** depende funcionalmente de **X** o que **X** determina o implica a **Y**, que se representa por  $X \rightarrow Y$ , si y solo si, cada valor de **X** tiene asociado en todo momento un único valor de **Y**.

**Dependencia funcional completa:** Si el descriptor X es compuesto, es decir, X(X1, X2), se dice que Y tiene *dependencia funcional completa* de X, si depende funcionalmente de X, pero no depende de ningún subconjunto de X.

**Dependencia funcional transitiva:** Sea la relación **R( X,Y,Z )**, en la que existen las siguientes dependencias funcionales: **X → Y**, y **Y → Z**, se dice que **Z** tiene dependencia transitiva respecto a **X**, a través de **Y**.

## Llave Primaria y Dependencia Funcional

Es necesario definir una llave primaria para cada tabla, que determine de forma única cada tupla. Además, para toda tabla se debe cumplir que toda columna que no es parte de la llave depende funcionalmente de la llave.

## Primera Forma Normal

Una tabla está en 1FN si todos los dominios de columnas contienen sólo valores atómicos o escalares.

### EMP

NEmp	NomEmp	NJefe	NomJefe	NDpto	NomDpto	NProj	NomProj	FecIni
7369	smith	7902	ford	20	Investigación	15	Factibilidad	10/05/2001
						35	Pruebas	20/05/2001
						45	Control	20/06/2001
7499	allen	7698	blake	30	Ventas	15	Factibilidad	05/05/2001
						25	Análisis	15/05/2001
						45	Control	20/06/2001

La tabla EMP no está en 1FN debido a las columnas NPROJ, NOMPROJ y FECINI.

La solución es “sacar” las columnas que dan problema y ponerlas en otra tabla acompañadas de su determinante:

### EMP

NEmp	NomEmp	NJefe	NomJefe	NDpto	NomDpto
7369	smith	7902	Ford	20	Investigación
7499	allen	7698	Blake	30	Ventas

### ASIGNACIÓN

NEmp	NProj	NomProj	FecIni
7369	15	Factibilidad	10/05/2001
7369	35	Pruebas	20/05/2001
7369	45	Control	20/06/2001
7499	15	Factibilidad	05/05/2001
7499	25	Análisis	15/05/2001
7499	45	Control	20/06/2001

## Segunda Forma Normal

Una tabla está en 2FN si está en 1FN y además se cumple que toda columna que no es parte de la llave de la tabla depende funcionalmente de toda la llave.

En la tabla ASIGNACIÓN las dependencias funcionales son:

NPROJ → NOMBPROJ  
(NEMP, NPROJ) → FECINI

En la tabla ASIGNACIÓN la llave es (NEMP, NPROJ) por lo tanto la tabla no está en 2FN, ya que, existe una columna que no es parte de la llave NOMBPROJ y que no depende de toda la llave.

La solución es “sacar” las columnas que dan problema y ponerlas en otra tabla acompañadas de su determinante.

### ASIGNACIÓN

NEmp	NProj	FecIni
7369	15	10/05/2001
7369	35	20/05/2001
7369	45	20/06/2001
7499	15	05/05/2001
7499	25	15/05/2001
7499	45	20/06/2001

### PROYECTOS

NProj	NomProj
15	Factibilidad
25	Análisis
35	Pruebas
45	Control

## Tercera Forma Normal

Una tabla está en 3FN si está en 2FN y además se cumple que toda columna que no es parte de la llave de la tabla depende sólo de la llave (dependencia transitiva)

### EMP

NEmp	NomEmp	NJefe	NomJefe	NDpto	NomDpto
7369	smith	7902	ford	20	Investigación
7499	allen	7698	blake	30	Ventas

Las dependencias funcionales son :

NEMP → NOMEMP  
 NJEFE → NOMJEFE  
 NDPTO → NOMDPTO  
 NEMP → NJEFE  
 NEMP → NDPTO  
 NEMP → NOMJEFE  
 NEMP → NOMDPTO

La tabla está en 2FN, ya que, todas las columnas que no es parte de la llave dependen funcionalmente de toda la llave, es decir,

$(NEMP) \rightarrow (NOMEMP, NJEFE, NOMJEFE, NDPTO, NOMDPTO)$

Pero no está en 3FN debido, por ejemplo, a que NOMDPTO además de depender de NEMP (que es parte de la llave) depende de NDPTO (que no es parte de la llave). En este caso, hablamos de que existe una dependencia transitiva.

La solución es “sacar” las columnas que dan problemas y ponerlas en otra tabla acompañadas de su determinante.

Primero se soluciona el problema de la columna NOMDPTO:

#### EMP

NEmp	NomEmp	NJefe	NomJefe	NDpto	NomDpto
7369	smith	7902	ford	20	Investigación
7499	allen	7698	blake	30	Ventas

#### DEPT

NDpto	NomDpto
20	Investigación
30	Ventas

Luego se soluciona el problema con la columna NOMJEFE

#### EMP

NEmp	NomEmp	NJefe	NDpto
7369	smith	7902	20
7499	allen	7698	30

Al proceso de “sacar” las columnas que dan problemas y ponerlas en otra tabla aparte junto con su determinante se le llama **descomposición**.

La descomposición de una tabla es correcta si al hacer el JOIN entre las tablas resultantes se vuelve a obtener la tabla original. Se le llama descomposición sin pérdida de información.

Al “llevarnos” el determinante de las columnas que se están sacando de la tabla, nos aseguramos que la descomposición sea sin pérdida de información.

## Ejercicio

El centro de computación de la Facultad posee un grupo de Consultoría dedicado a la atención de usuarios. Dicho grupo está formado por alumnos regulares de la Facultad, los que tienen dominio de algún software específico. Para cada consultor existe un horario de atención preestablecido y el propio jefe del grupo está a cargo de controlar el funcionamiento del servicio.

Se pide normalizar el modelo relacional hasta la 3FN. Para ello determine claramente todas las dependencias funcionales existentes. Es importante que justifique todos sus pasos.

Una vez que haya normalizado el modelo relacional, infiera el modelo entidad-relación, identificando todas sus componentes.

horario				
matricula	num_dia	cod_mod	hora_inicio	hora_fin
2039	1	1	08,50	10,00
2039	2	2	10,25	11,75
2020	3	1	08,50	10,00

asistencia		
fecha	matricula	cod_mod
24/11/98	2039	1
25/11/98	2029	2
25/11/98	2020	1

consultor					
cod_dpto	Nom_dpto	nombre	matricula	cod_soft	descripcion
10	Computación	Alfredo	2039	0001	Excel 5,0
				0002	Word 6,0
				0003	Pascal
				0004	Turing
20	Matemáticas	Claudia	2020	0005	C
				0006	PL/SQL
				0003	Pascal

## Solución

La tabla *asistencia* se encuentra en 3FN, ya que estando la llave primaria compuesta por los campos (*fecha,matricula*) se tiene la siguiente dependencia funcional:

$$(fecha,matricula) \rightarrow cod\_mod$$

lo cual indica que todo campo que no pertenece a la llave primaria (en este caso solamente *cod\_mod*) depende SOLO de la llave primaria.

La tabla *horario* y *consultor*, no están en ninguna forma normal ya que ambas presentan columnas con valores vectoriales. (no atómicos)

En la tabla *horario*, estas columnas son *hora\_inicio, hora\_fin* por lo tanto hay que llevarlas a otra tabla junto con su determinante que es *cod\_mod*. Esto se traduce en:

horario_diario		
Matricula*	num_dia	cod_mod
2039	1	1
2039	2	2
2020	3	1

modulo		
cod_mod*	hora_inicio*	hora_fin*
1	08,50	10,00
2	10,25	11,75
1	08,50	10,00

De esta manera ambas tablas quedan en 3FN, siendo sus dependencias funcionales las siguientes:

*horario\_diario*:

Para soportar el caso de que un alumno pueda tener dos módulos distintos un mismo día, es necesario que la llave primaria esté formada por las tres columnas (*matricula,num\_dia,cod\_mod*).

*modulo*:

Siendo *cod\_mod* la llave primaria, se tienen las dependencias:

$$cod\_mod \rightarrow fecha\_inicio$$

$$cod\_mod \rightarrow fecha\_fin$$

Para la tabla *consultor*, primero debemos llevar a otra tabla la columna *descripción* con su determinante *cod\_soft*:

consultor				
cod_dpto	nom_dpto	nombre	matricula*	cod_soft
10	Computación	Alfredo	2039	0001
				0002
				0003
				0004
20	Matemáticas	Claudia	2020	0005
				0006
				0003

software	
cod_soft*	descripcion
0001	Excel 5,0
0002	Word 6,0
0003	Pascal
0004	Turing
0005	C
0006	PL/SQL
0003	Pascal



Así, la tabla *software* queda en 3FN. Pero la tabla *consultor* sigue sin ninguna forma normal, pues la columna *cod\_soft* tiene datos repetidos. Luego se debe separar en otra tabla junto con su determinante que es *matricula*.

consultor			
cod_dpto	nom_dpto	nombre	matricula*
10	Computación	Alfredo	2039
20	Matemáticas	Claudia	2020

conocimiento	
matricula*	cod_soft*
2039	0001
2039	0002
2039	0003
2039	0004
2020	0005
2020	0006
2020	0003

La tabla *conocimiento* está en 3FN si la llave primaria está formada por ambos campos (*matricula, cod\_soft*).

Las dependencias funcionales de la tabla *consultor* si la llave primaria es *matricula*, son:

$matricula \rightarrow nombre$   
 $matricula \rightarrow nom\_dpto$   
 $matricula \rightarrow cod\_dpto$   
 $cod\_dpto \rightarrow nom\_dpto$

esto nos dice que la tabla está en 2FN (todo campo que no pertenece a la llave primaria, depende funcionalmente de TODA la llave primaria), pero no está en 3FN ya que, *nom\_dpto* además de depender de la llave primaria depende de *cod\_dpto*. Luego la separamos en otra tabla junto con su determinante.

consultor		
matricula*	nombre	cod_dpto
2039	Alfredo	10
2020	Claudia	20

departamento	
cod_dpto*	nom_dpto
10	Computación
20	Matemáticas

De esta forma todo queda en 3FN. Las dependencias funcionales de estas tablas son:

*consultor:*       $matricula \rightarrow nombre$       *departamento:*       $cod\_dpto \rightarrow nom\_dpto$   
 $matricula \rightarrow cod\_dpto$

El modelo entidad – relación queda de la siguiente forma:

