

Tema 5. Normalización

Contenido

1 Problemas del esquema relacional	2
2 Formas normales	2
3 Primera Forma Normal. 1FN	3
4 Dependencias funcionales	5
4.1 Dependencia funcional.....	5
4.2 Dependencia funcional completa o total	5
4.3 Dependencia funcional transitiva	5
5 Segunda forma normal. 2FN	6
6 Tercera forma normal. 3FN	7
7 Forma normal de boyce-codd. FNBC o BCFN.	8

1 Problemas del esquema relacional

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 que una base de datos tenga los siguientes problemas:

- **Redundancia.** Se llama así a los datos que se repiten continuamente e innecesariamente por las tablas de las bases de datos. Cuando es excesiva es evidente que el diseño hay que revisarlo, es el primer síntoma de problemas y se detecta fácilmente.
- **Ambigüedades.** Datos que no clarifican suficientemente el registro al que representan. Los datos de cada registro podrían referirse a más de un registro o incluso puede ser imposible saber a qué ejemplar exactamente se están refiriendo. Es un problema muy grave y difícil de detectar.
- **Pérdida de restricciones de integridad.** Normalmente debido a dependencias funcionales. Más adelante se explica este problema. Se arreglan fácilmente siguiendo una serie de pasos concretos.
- **Anomalías en operaciones de modificación de datos.** El hecho de que al insertar un solo elemento haya que repetir tuplas en una tabla para variar unos pocos datos. O que eliminar un elemento suponga eliminar varias tuplas necesariamente (por ejemplo que eliminar un cliente suponga borrar seis o siete filas de la tabla de clientes, sería un error muy grave y por lo tanto un diseño terrible).

El principio fundamental reside en que las tablas deben referirse a objetos o situaciones muy concretas, relacionados exactamente con elementos reconocibles por el sistema de información de forma inequívoca. Cada fila de una tabla representa inequívocamente un elemento reconocible en el sistema. Lo que ocurre es que conceptualmente es difícil agrupar esos elementos correctamente.

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 pueden resolver usando reglas de normalización. Estas reglas suelen forzar la división de una tabla en dos o más tablas para arreglar ese problema.

2 Formas normales

La idea inicial es transformar cada relación del modelo lógico en una tabla en el modelo relacional. Pero hay casos en los que esta regla tiene matices y no se cumple.

Las formas normales se corresponden a una teoría de normalización iniciada por el propio Codd y continuada por otros autores (entre los que destacan Boyce y Fagin). Codd definió en 1970 la primera forma normal, desde ese momento aparecieron la segunda, tercera, la Boyce-Codd, la cuarta y la quinta forma normal.

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).

La teoría de formas normales es una teoría absolutamente matemática, pero en este tema se describen de forma más intuitiva.

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.

La forma normal de Boyce-Codd, la cuarta y la quinta forma normal no van a ser materia del curso. Aunque la primera se explica al final de este tema.

3 Primera Forma Normal 1FN

Es una forma normal inherente al esquema relacional. Es decir toda tabla realmente relacional la cumple.

Se dice que una tabla se encuentra en primera forma normal, 1FN, si impide que un atributo de una tupla pueda tomar más de un valor. La tabla:

TRABAJADOR		
<u>DNI</u>	Nombre	Departamento
12121212A	Andrés	Mantenimiento
12345345G	Andrea	Dirección Gestión

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 pasar a primera forma normal hay que dividir la tabla anterior en dos (los atributos que forman la clave primaria están subrayados):

TRABAJADOR	
<u>DNI</u>	Nombre
12121212A	Andrés
12345345G	Andrea

<u>DNI</u>	Departamento
12121212A	Mantenimiento
12345345G	Dirección
12345345G	Gestión

Estas tablas sí están en primera forma normal.

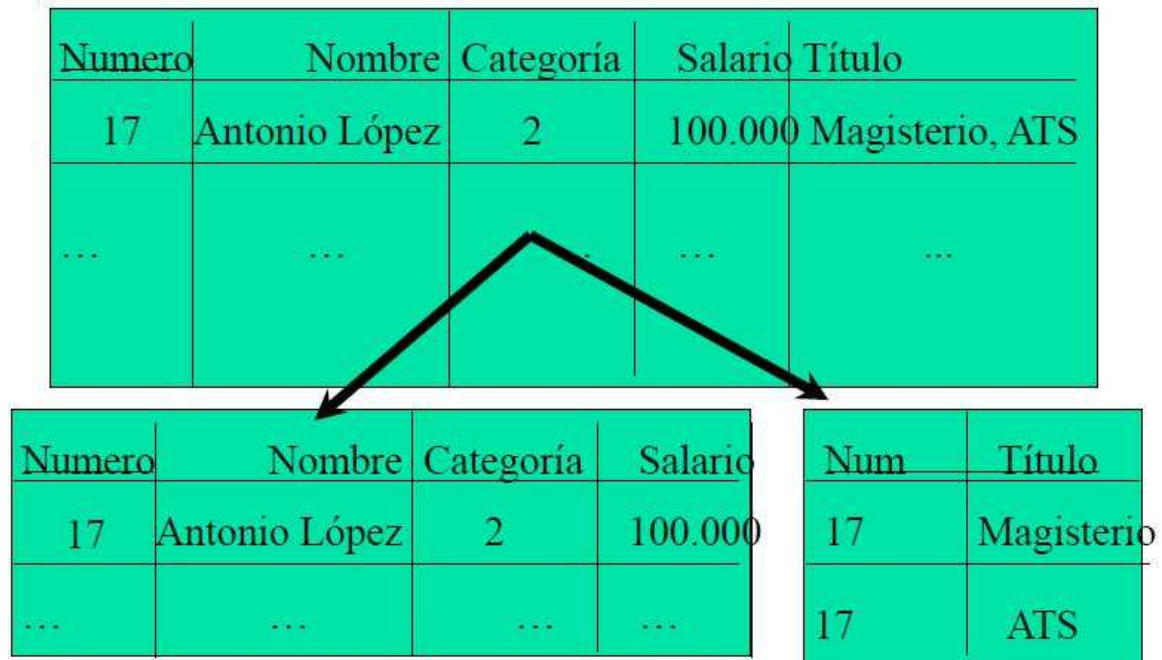
Pasos a seguir para pasar las tablas a 1FN:

Se divide la relación en dos:

- Se localizan los atributos que forman la clave principal.
- Se descompone la tabla realizando dos proyecciones:
- Relación 1: Clave + atributos únicos: La clave con los atributos que tiene valores únicos, tomando la tabla nueva el nombre de la tabla original.
- Relación 2: Clave + atributos múltiples. La **nueva clave** está formada por ambos campos. Se crea otra tabla con la clave y los atributos que tienen valores múltiples, distribuyendo cada valor en una fila, con lo que

cada fila tendrá un valor elemental para el atributo. La tabla que se genera tendrá un nombre mnemotécnico compuesto por la abreviatura de los atributos que la definen.

Ejemplo:



4 Dependencias funcionales

4.1 Dependencia funcional

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 $X \rightarrow Y$.

Por ejemplo el nombre de una persona depende funcionalmente del DNI; es decir para un DNI concreto sólo hay un nombre posible. En la tabla del ejemplo anterior, el departamento no tiene dependencia funcional, ya que para un mismo DNI puede haber más de un departamento posible. Pero el nombre sí que depende del DNI.

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

Cada dependencia funcional es una clase especial de regla de integridad y representa una relación de uno a muchos.

4.2 Dependencia funcional completa o total

Un conjunto de atributos (Y) tiene una dependencia funcional completa sobre otro conjunto de atributos (X) si Y tiene dependencia funcional de X y además no se puede obtener de X un conjunto de atributos más pequeño que consiga una dependencia funcional de Y (es decir, no hay en X un determinante formado por un número menor de atributos).

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 completa ya que el dni individualmente, también produce una dependencia funcional sobre apellidos. El dni sí produce una dependencia funcional completa sobre el campo apellidos.

$(\text{DNI, EMPRESA}) \rightarrow \text{NOMBRE}$ no es total puesto que Nombre depende únicamente del Dni. A esta dependencia se la denomina parcial.

La dependencia funcional total sería $(\text{DNI, EMPRESA}) \rightarrow \text{SUELDO}$

Las dependencias que interesan para tratar las anomalías y su solución son las dependencias funcionales totales.

Una dependencia funcional completa se denota como $X \Rightarrow Y$.

4.3 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 Z.

Entonces ocurre que X produce una dependencia funcional transitiva sobre Z.

Esto se denota como: $(X \rightarrow Y) \wedge (Y \rightarrow Z) \Rightarrow X \rightarrow Z$.

Por ejemplo, supongamos que un profesor puede ser tutor de varios grupos, si X es el atributo **Código de Grupo de un instituto**, e Y es el atributo **Código Tutor**. Entonces $X \rightarrow Y$ (el tutor depende funcionalmente del código de grupo).

$\text{Código de grupo} \rightarrow \text{Código Tutor}$

Si Z representa el **Código del departamento**, entonces $Y \rightarrow Z$ (el código del departamento depende funcionalmente del código tutor, cada tutor sólo puede estar en un departamento). Como ocurre que X no depende funcionalmente de Y (el **código del grupo** no depende funcionalmente del **código tutor** porque un **código tutor** se puede corresponder con varios **códigos de grupo**), entonces $X \not\rightarrow Z$ (el código del departamento depende transitivamente del código del grupo).

Código Tutor \rightarrow Código del departamento

Código de grupo \rightarrow Código del departamento

5 Segunda forma normal 2FN

La segunda forma normal trata de relaciones entre los atributos clave y los no clave.

Una tabla está en segunda forma normal, 2FN, si está en 1FN y además cada atributo que no forma parte de la clave primaria depende completamente de la clave primaria de la tabla (la comprobación sólo hay que hacerla cuando la clave primaria está compuesta por varios atributos).

Pasos a seguir para pasar las tablas a 2FN:

- Se comprueba que todas estén en 1FN.
- Se crea una primera tabla con la clave de la inicial y todos los atributos que tienen una dependencia funcional total con ella.
- Se crea una segunda tabla para los atributos que no dependan de la totalidad de la clave compuesta.
- El identificador o clave primaria de la nueva tabla será la parte de la clave compuesta de que dependen el atributo o atributos seleccionados.

Ejemplo:

ALUMNOS				
DNI	Cod Curso	Nombre	Apellido1	Nota
12121219A	34	Pedro	Valiente	9
12121219A	25	Pedro	Valiente	8
3457775G	34	Ana	Fernández	6
5674378J	25	Sara	Crespo	7
5674378J	34	Sara	Crespo	6

Suponiendo que el DNI y el código de curso formen una clave principal para esta tabla, sólo la nota tiene dependencia funcional completa. El nombre y los apellidos dependen de forma completa del DNI. La tabla no está en 2FN, para arreglarlo:

ALUMNOS		
DNI	Nombre	Apellido1
12121219A	Pedro	Valiente
3457775G	Ana	Fernández
5674378J	Sara	Crespo

ASISTENCIA		
DNI	Cod Curso	Nota
12121219A	34	9
12121219A	25	8
3457775G	34	6
5674378J	25	7
5674378J	34	6

6 Tercera forma normal 3FN

Ocurre cuando una tabla está en 2FN y todo atributo que no forma parte de la clave primaria depende de la clave de forma no transitiva, es decir, no hay dependencia funcional entre dos atributos no clave.

Pasos a seguir para que una tabla esté en 3FN:

- Se comprueba que está en 1FN y en 2FN.
- Por cada atributo de una tabla que dependa de otro que no sea identificador, se crea una nueva tabla (si no existe) a la que se le pasan esos atributos y por clave el atributo del cual dependían. Este atributo se queda en la primera tabla como clave ajena. Si la nueva tabla sigue sin estar en 3FN se repite el proceso de nuevo hasta que obtengamos una 3FN.

Ejemplo:

ALUMNOS				
DNI	Nombre	Apellido1	Cod Provincia	Provincia
12121349A	Salvador	Velasco	34	Palencia
12121219A	Pedro	Valiente	34	Palencia
3457775G	Ana	Fernández	47	Valladolid
5674378J	Sara	Crespo	47	Valladolid
3456858S	Marina	Serrat	08	Barcelona

ALUMNOS			
DNI	Nombre	Apellido1	Cod Provincia
12121349A	Salvador	Velasco	34
12121219A	Pedro	Valiente	34
3457775G	Ana	Fernández	47
5674378J	Sara	Crespo	47
3456858S	Marina	Serrat	08

PROVINCIA	
Cod Provincia	Provincia
34	Palencia
47	Valladolid
08	Barcelona

7 Forma normal de Boyce-Codd. FNBC o BCFN.

Ocurre si una tabla está en tercera forma normal y además todo determinante es una clave candidata. Ejemplo:

Consideremos una empresa donde un trabajador puede trabajar en varios departamentos. En cada departamento hay varios responsables, pero cada trabajador sólo tiene asignado uno. Tendríamos una tabla con las columnas:

Trabajador, Departamento, Responsable

La única clave candidata es #Trabajador, #Departamento (que será por tanto la clave primaria).

Si añadimos la limitación de que el trabajador que sea responsable de un departamento sólo puede serlo de uno, este detalle produce una dependencia funcional ya que: **Responsable → Departamento**

Por lo tanto hemos encontrado un determinante (Responsable) que sin embargo no es clave candidata. Por ello, esta tabla no está en FNBC. En este caso la redundancia ocurre por mala selección de clave. La repetición del par [Departamento + Responsable] es innecesaria y evitable

ORGANIZACIÓN		
#Trabajador	#Departamento	Responsable
1	Producción	10
2	Producción	11
3	Ventas	12
3	Producción	10
4	Producción	13
5	Ventas	14
5	Producción	11
6	Ventas	12
7	Ventas	14

La solución sería:

PERSONAL	
#Trabajador	#Responsable
1	10
2	11
3	12
3	10
4	13
5	14
5	11
6	12
7	14

RESPONSABLES	
#Responsable	Departamento
10	Producción
11	Producción
12	Ventas
13	Producción
14	Ventas

En las formas de Boyce-Codd hay que tener cuidado al descomponer la tabla de partida en nuevas tablas ya que se podría perder información si no se hace bien.