

Tema 2.Bases de datos relacionales.

NORMALIZACIÓN

Índice.

1. Problemas del modelo relacional.
2. Formas Normales.
3. Primera Forma Normal (1FN).
 1. Dependencias funcionales.
 2. Dependencia funcional completa.
 3. Dependencia funcional elemental.
 4. Dependencia funcional transitiva.
4. Segunda forma normal (2FN).
5. Tercera forma normal (3FN).
6. Forma normal de Boyce-Codd

1. Problemas del modelo relacional.

Normalmente, al construir el modelo relacional a partir del diagrama E/R tendríamos una buena base de datos. En ocasiones, debido a fallos del diseño u otro tipo de problemas, la base de datos puede incorporar problemas como:

REDUNDANCIA: Se produce cuando **los datos se repiten continua e innecesariamente** por las tablas de la base de datos. Cuando es excesiva es evidente que el diseño hay que revisarlo.

AMBIGÜEDADES: Se produce cuando **los datos no clarifican suficientemente a la tabla a la que pertenecen**. Los datos podrían referirse a más de una tabla incluso no saber a qué tabla se refieren. (es un problema difícil de detectar).

PERDIDA DE RESTRICCIONES DE IDENTIDAD:

Las restricciones son estructuras u ocurrencias no permitidas. Existen dos tipos de restricciones:

- Restricciones Inherentes.
- Restricciones de usuario.

1. Problemas del modelo relacional.

Restricciones Inherentes.

Son una serie de reglas impuestas por el modelo, como:

- No puede haber 2 tuplas iguales.
- El orden de las tuplas no es significativo.(filas)
- El orden de los atributos no es significativo (columnas).
- Cada atributo solo puede tomar un único valor.
- La regla de integridad de entidad establece que:
“Ningún atributo que forme parte de la clave puede tomar valor nulo, es decir, un valor desconocido o inexistente”.

1. Problemas del modelo relacional.

Restricciones de usuario.

Son reglas que vienen impuestas por el usuario al describir el modelo relacional.

La regla de integridad referencial establece que:

“Los valores de una clave ajena o bien coinciden con los de la clave a la que refieren o son nulos”.

Ejemplo:

SUCURSAL (Num_sucursal#,Activo,Ciudad)

CUENTA (Num_cuenta#,Saldo,Num_sucursal)

Las cuentas de nuestra base de datos deberán pertenecer a una sucursal existente o si se desconoce la sucursal, no se tendrá ningún valor para ese atributo.

1. Problemas del modelo relacional.

ANOMALIAS EN OPERACIONES DE MODIFICACIÓN DE DATOS.

Un ejemplo seria si queremos insertar un solo elemento y tenemos que repetir filas en una tabla, o bien si deseamos eliminar un solo elemento y para ello debemos eliminar varias filas.

Ejemplo:

Si tenemos la tabla CLIENTES y para eliminar un solo cliente debemos eliminar de la misma tabla 5 o 6 filas, seria un error muy grave y por tanto un diseño terrible.

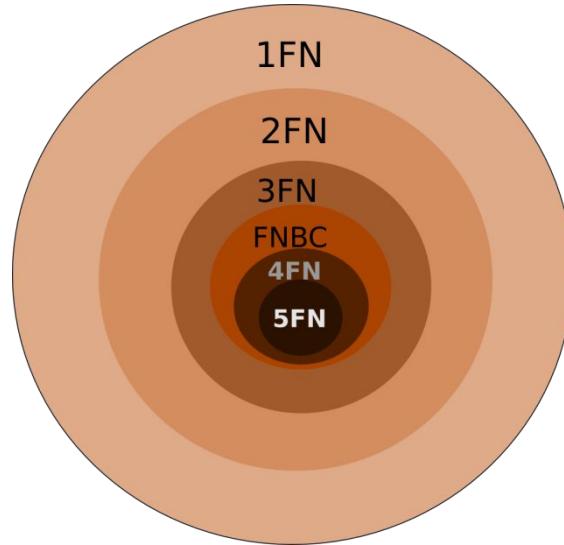
1. Problemas del modelo relacional.

- Estos problemas se **agravan** si no se sigue el modelo conceptual y se crea directamente el modelo relacional. En ese caso el diseño tiene una garantía casi asegurada de funcionar mal.
- Cuando aparecen estos problemas se pueden resolver usando reglas de **normalización**. Estas reglas suelen forzar la división de una tabla en dos o más para arreglar el problema siendo estas de menor tamaño (con menor número de campos).
- Al aumentar el número de tablas se suele decir que se penaliza por recuperar información.

2. FORMAS NORMALES.

Las formas normales se corresponden a la teoría de normalización iniciada por Codd y continuada por otros autores (como Boyce y Fagin). Codd en 1970 definió la 1FN, desde ese momento aparecieron la segunda, la tercera, la Boyce-Codd, la cuarta y la quinta forma normal.

Una tabla puede estar en 1FN y no en 2FN, pero al revés no. Es decir, cada FN debe cumplir las anteriores.



Los números más altos de FN son más restrictivos.

2. FORMAS NORMALES.

- La teoría de normalización es absolutamente matemática basada en el concepto de DEPENDENCIA FUNCIONAL.
- Hay muchos autores que opinan que basta con llegar a la forma de Boyce-Codd, ya que la cuarta y, sobre todo, la quinta son polémicas. 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.

2.3. PRIMERA FORMA NORMAL (1FN).

Se dice que una tabla **se encuentra en primera forma normal** si impide que un atributo de una tupla (fila) puede tomar más de un valor. Los atributos deber ser **atómicos** (1 solo valor).

TRABAJADOR		
DNI	NOMBRE	DEPARTAMENTO
12121212A	Andrés	Mantenimiento
12345345G	Andrea	Dirección Gestión

Para pasar a 1FN hacemos lo siguiente

TRABAJADOR		
DNI	NOMBRE	DEPARTAMENTO
12121212A	Andrés	Mantenimiento
12345345G	Andrea	Dirección
12345345G	Andrea	Gestión

2.4. DEPENDENCIAS FUNCIONALES

A)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 representa:

X->Y

Al conjunto X se llama DETERMINANTE.

Al conjunto Y se llama IMPLICADO.

Ejemplos:

Nombre depende funcionalmente del **DNI** (DNI ->Nombre)

Para cada DNI hay solo un nombre posible

Departamento no depende funcionalmente de **DNI** (DNI ->Departamento)

Para cada DNI puede haber más de un departamento.

Ej: La persona con DNI 123456345G puede estar en el departamento de gestión y de administración.

2.4. DEPENDENCIAS FUNCIONALES

B)DEPENDENCIA FUNCIONAL COMPLETA.

Un conjunto de atributos (Y) tiene una dependencia funcional completa de 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, **depende funcionalmente de X pero no depende de ningún subconjunto de X**).

Si $X(X_1, X_2)$ (X está formado por dos atributos o conjunto de atributos)

$$X \rightarrow Y$$

$$X_1 \not\rightarrow Y$$

$$X_2 \not\rightarrow Y$$

No depende de ningún subconjunto de X

Una dependencia funcional completa se denota con: **$X \Rightarrow Y$**

2.4. DEPENDENCIAS FUNCIONALES

Ejemplo 1.

Supongamos la tabla CLIENTES(DNI,Nombre,Apellido)

Si $X(DNI,Nombre) \rightarrow Y(Apellido)$

Se tiene que

$X \rightarrow Y$ (Y depende funcionalmente de X, para cada valor de Y, hay un solo valor de X)

Pero

$X \Rightarrow Y$ (Y no tiene dependencia funcional completa sobre X, ya que el DNI por si solo produce dependencia funcional sobre Y. $DNI \rightarrow Apellido$)

2.4. DEPENDENCIAS FUNCIONALES

Ejemplo 2.

Supongamos una empresa en la que un empleado puede trabajar en varios proyectos, realizando una sola función en cada uno de ellos (consultor, analista, programador, etc.) aunque puede ser distinta según el proyecto. Tendríamos que:

(DNI_Empleado,Cod_Proyecto)=>Función.

Sería dependencia funcional completa ya que:

DNI_Empleado no determina la función (también depende del proyecto)

COD_Proyecto no determina la función (también depende del empleado)

2.4. DEPENDENCIAS FUNCIONALES

C) DEPENDENCIA FUNCIONAL ELEMENTAL

Se produce cuando X e Y forman una dependencia funcional completa y además Y es un único atributo.

D) DEPENDENCIA FUNCIONAL TRANSITIVA.

Se produce cuando hay tres conjuntos de atributos X,Y,Z.

Si se tiene que:

- $X \rightarrow Y$ Y depende funcionalmente de X
- $Y \rightarrow Z$ Z depende funcionalmente de Y
- $Y \not\rightarrow X$ X no depende funcionalmente de Y

Entonces se dice que existe DEPENDENCIA FUNCIONAL TRANSITIVA. Se denota por

$X \rightarrow\!\!\!-> Z$

2.4. DEPENDENCIAS FUNCIONALES

Ejemplo.

Supongamos una empresa en la que cada empleado trabaja para un único departamento. Supongamos que se dan las siguientes dependencias.

Cod_Empleado-> Cod_Depto (Para cada empleado existe un único código de departamento)

Cod_Depto -> Nombre_Depto (Para cada código de departamento existe un único nombre de departamento)

Además Cod_Depto $\not\rightarrow$ Cod_Empleado

Entonces se da que:

Cod_Empleado -->Nombre_Depto

El nombre de departamento depende transitivamente del Código de empleado.

2.5. SEGUNDA FORMA NORMAL (2FN)

Una tabla está en **segunda forma normal** si:

- **Está en 1FN.**
- **Cada atributo que no sea clave depende de forma funcional completa respecto de la clave.**

Esto es: toda la clave debe hacer dependientes al resto de los atributos, si hay atributos que dependen de solo parte de la clave entonces esa parte de la clave y esos atributos formaran la tabla.

2.5. SEGUNDA FORMA NORMAL (2FN)

Ejemplo:

ALUMNOS				
<u>DNI#</u>	<u>Cod_curso#</u>	Nombre	Apellido1	Nota
12121212A	34	Pedro	Valiente	9
12121212A	25	Pedro	Valiente	8
34343434G	34	Ana	Fernández	6
56785678J	25	Sara	Crespo	7
56785678J	34	Sara	Crespo	6

Clave= DNI y Cod_curso

Solo la Nota tiene dependencia funcional completa de la clave. El Nombre y Apellido1 dependen de forma completa del DNI.

La tabla no estaría en 2FN. Para arreglarlo formamos una tabla con los atributos y parte de la clave.

2.5. SEGUNDA FORMA NORMAL (2FN)

ALUMNOS				
<u>DNI#</u>	<u>Cod_curso#</u>	Nombre	Apellido1	Nota
12121212A	34	Pedro	Valiente	9
12121212A	25	Pedro	Valiente	8
34343434G	34	Ana	Fernández	6
56785678J	25	Sara	Crespo	7
56785678J	34	Sara	Crespo	6

CALIFICACIONES		
<u>DNI#</u>	<u>Cod_curso#</u>	Nota
12121212A	34	9
12121212A	25	8
34343434G	34	6
56785678J	25	7
56785678J	34	6

ALUMNOS		
<u>DNI#</u>	Nombre	Apellido1
12121212A	Pedro	Valiente
34343434G	Ana	Fernández
56785678J	Sara	Crespo

Aquí las dos tablas sí estarían en 2FN

2.6. TERCERA FORMA NORMAL (3FN)

Una tabla está en 3FN si:

- **Está en 2FN.**
- **Ningún atributo que no sea clave depende transitivamente de la clave de la tabla.**

Es decir, no estaría en 3FN cuando algún atributo depende funcionalmente de atributos que no son clave.

Ejemplo:

ALUMNOS				
DNI#	Nombre#	Apellido1	Cod_Provincia	Provincia
12121212A	Salvador	Velasco	34	Palencia
23232323B	Pedro	Valiente	34	Palencia
34343434G	Ana	Fernández	47	Valladolid
56785678J	Sara	Crespo	47	Valladolid
12341234P	Marina	Serrat	08	Barcelona

La provincia depende funcionalmente del código de provincia, por lo que la tabla no estaría en 3FN. El arreglo sería el siguiente:

2.6. TERCERA FORMA NORMAL (3FN)

ALUMNOS				
<u>DNI#</u>	<u>Nombre#</u>	Apellido1	Cod_Provincia	Provincia
12121212A	Salvador	Velasco	34	Palencia
23232323B	Pedro	Valiente	34	Palencia
34343434G	Ana	Fernández	47	Valladolid
56785678J	Sara	Crespo	47	Valladolid
12341234P	Marina	Serrat	08	Barcelona

ALUMNOS				PROVINCIA	
<u>DNI#</u>	<u>Nombre#</u>	Apellido1	Cod_Provincia	Cod_Provincia	Provincia
12121212A	Salvador	Velasco	34	34	Palencia
23232323B	Pedro	Valiente	34	47	Valladolid
34343434G	Ana	Fernández	47	08	Barcelona
56785678J	Sara	Crespo	47		
12341234P	Marina	Serrat	08		

Aquí las dos tablas sí estarían en 3FN

2.7. FORMA NORMAL DE BOYCE-COD (FNBC)

Una tabla está en FNBC si

- **Está en 3FN.**
- **Todo determinante es clave.**

Ejemplo 1:

Supongamos en un centro de secundaria, para cada alumno y asignatura tenemos asignado un profesor/a.

Está en 3FN (no hay dependencias transitivas)

(DNI, Asignatura)-> Profesor

Para cada alumno y asignatura hay un único profesor.

Pero también se tiene que

Profesor ->Asignatura

Para cada profesor hay una única asignatura.

Profesor sería determinante, pero no forma parte de la clave, por lo que no estaría en FNBC.

PROFESORES		
<u>DNI#</u>	<u>Asignatura#</u>	<u>Profesor/a</u>
12121212A	Lengua	Eva
12121212A	Matemáticas	Andrés
34534555G	Lengua	Eva
56756788J	Matemáticas	Guillermo
56756788J	Lengua	Julia
56345634H	Matemáticas	Guillermo

2.7. FORMA NORMAL DE BOYCE-COD (FNBC)

Para solucionar el problema elegimos el DNI y profesor como clave y dividimos la tabla. Quedaría:

PROFESORES	
<u>DNI#</u>	<u>Profesor/a#</u>
12121212A	Eva
12121212A	Andrés
34534555G	Eva
56756788J	Guillermo
56756788J	Julia
56345634H	Guillermo

ASIGNATURA_PROFESOR	
<u>Asignatura#</u>	<u>Profesor/a#</u>
Lengua	Eva
Matemáticas	Andrés
Matemáticas	Guillermo
Lengua	Julia

2.7. FORMA NORMAL DE BOYCE-COD (FNBC)

Ejemplo 2.

Supongamos la siguiente tabla

VENTAS(Cod_pieza,Cod_almacen,Nom_almacen,Cantidad)

Donde los almacenes se determinan únicamente tanto por el código como por el almacén. Podemos elegir 2 claves.

A)(Cod_pieza,Cod_almacen)

B)(Cod_pieza,Nom_almacen)

2.7. FORMA NORMAL DE BOYCE-COD (FNBC)

A) Si elegimos la primera opción.

VENTAS(Cod_pieza#,Cod_almacen#,Nom_almacen,Cantidad)

Tendríamos que

Nom_almacen-> Cod_almacen

Nom_almacen es un determinante que no es clave. No estaría en FNBC

B) Si elegimos la segunda opción.

VENTAS(Cod_pieza#, Nom_almacen#, Cod_almacen, Cantidad)

Tendríamos que

Cod_almacen-> Nom_almacen

Cod_almacen es un determinante que no es clave. No estaría en FNBC

2.7. FORMA NORMAL DE BOYCE-COD (FNBC)

Si queremos que cumpla la FNBC debemos descomponer la tabla. Por ejemplo de la siguiente forma:

ALMACENES(Cod_almacen#,Nom_almacen#)

VENTAS(Cod_pieza#,Cod_almacen#,Cantidad)

Está en FNBC porque todo determinante es clave.

Fin

Relación de ejercicios de normalización.