

Teoría Sección 07 – Normalización.

- ¿Que es una dependencia funcional?

Decimos que un atributo B es dependiente funcional de A (pueden ser varios atributos de A) si el valor de A determina un valor simple de B en todo momento.

La notación sería algo como:

$$\begin{array}{lcl} A & \longrightarrow & B \\ A_1, A_2, \dots, A_n & \longrightarrow & B \end{array}$$

Ejemplo:

studentID	Curso	Costo	NivelStud
100	Matematica	150	Avanzado
100	Fisica	175	Intermedio
150	Fisica	175	Basico
175	Quimica	120	Avanzado
175	Biologia	100	Intermedio
200	Matematica	150	Basico
210	Fisica	175	Avanzado

Curso \longrightarrow Costo

studentID, Curso \longrightarrow NivelStud

- ¿Que es una dependencia transitiva?

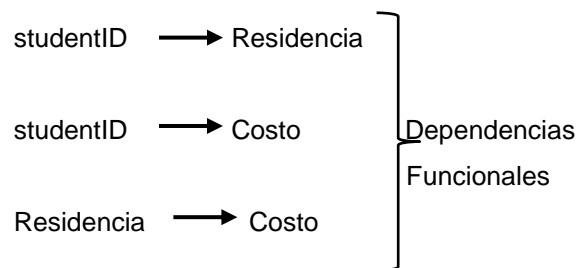
Decimos que existe una dependencia transitiva cuando hay una dependencia intermedia. Tomando 3 atributos A, B y C y se cumple que $A \rightarrow B$ y $B \rightarrow C$ podemos decir que la dependencia transitiva existe en la tabla.

La notación sería algo como:

$$A \rightarrow B \rightarrow C$$

Ejemplo:

studentID	Residencia	Costo
100	AlphaBeta	1100
110	Perkin	1200
120	Gatewarm	1150
130	Perkin	1200
140	Lionside	1220
150	AlphaBeta	1100
160	Lionside	1220



Como resultado tenemos la siguiente dependencia transitiva:

$$\text{studentId} \rightarrow \text{Residencia} \rightarrow \text{Costo}$$

- ¿Que es una normalización en BD?

Es una técnica que ayuda a desarrollar la estructura lógica de los datos de un sistema informático en el modelo relacional.

Fue introducida por E.F. Codd en 1972 donde define una estrategia de diseño de abajo hacia arriba, partiendo de los atributos y estos se van agrupando en tablas según su afinidad.

Los objetivos de la normalización son:

1. Controlar la redundancia de la información.
2. Evitar pérdidas de información.
3. Mantener la consistencia de los datos.
4. Capacidad para representar toda la información.

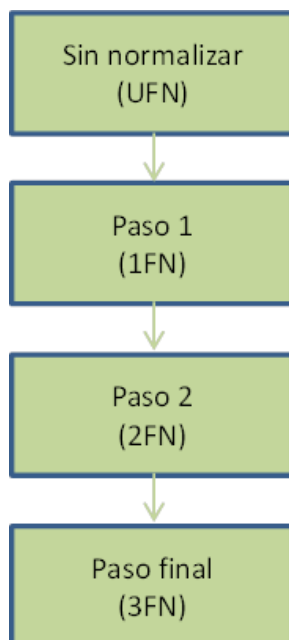
- ¿Cuáles son los pasos para realizar la normalización?

Al aplicar esta técnica nos debemos asegurar que cada relación o tabla cumple con ciertas reglas o condiciones.

Si alguna regla o condición no se cumple, entonces esa tabla se debe descomponer en varias relaciones que si cumplan con esa regla.

Normalmente, con realizar una normalización de 1era, 2da, y 3era forma es suficiente para tener un modelo estable y menos susceptible a las anomalías estudiadas en este curso.

En general, los pasos son:



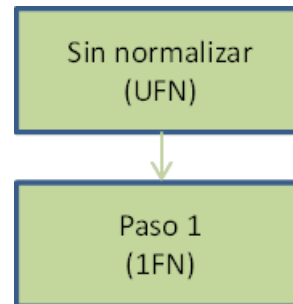
1. Descomponer todos los grupos de datos en atributos atómicos (1FN), además de que cada tabla debe poseer su PK.
2. Eliminar todas las relaciones en la que los datos no dependan completamente de la llave primaria de la tabla (2FN).
3. Eliminar todas las relaciones que contengan dependencias transitivas (3FN).

- 1era Forma Normal (1FN o 1NF).

Descomponer todos los grupos de datos en atributos atómicos, buscar redundancias y la tabla debe poseer, al menos, una clave primaria.

Chequeamos que nuestros datos no sean compuestos, algunos ejemplos comunes:

- Nombre de personas.
- Direcciones.
- Números de teléfonos.



Ejemplo:

idEmp	Nombre	Telefono
e0001	Pablo Perez	444-4444
e0002	Andrea Vargas	111-1111, 222-2222

Verificamos:

- PK 😊
- Datos atómicos ❌
- Redundancias 😊

idEmp	Nombre	Apellido	Telefono
e0001	Pablo	Perez	444-4444
e0002	Andrea	Vargas	111-1111
e0002	Andrea	Vargas	222-2222

Verificamos:

- PK 😊
- Datos atómicos 😊
- Redundancias ❌

idEmp	Nombre	Apellido	idEmp	Telefono	Verificamos:
e0001	Pablo	Perez	e0001	444-4444	
e0002	Andrea	Vargas	e0002	111-1111	
			e0002	222-2222	

Verificamos:

- PK 😊
- Datos atómicos 😊
- Redundancias 😊

- 2da Forma Normal (2FN o 2NF).

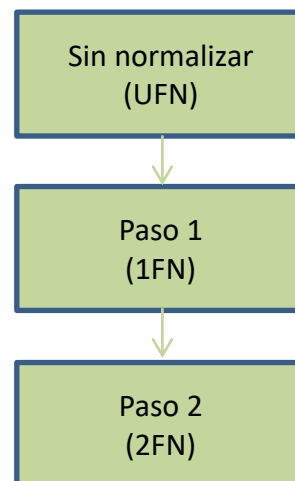
La tabla debe estar en 1FN y a su vez cada atributo no clave es total y funcionalmente dependiente de su clave primaria.

Debemos chequear las dependencias funcionales de la tabla.

A → B

A: clave primaria.

B: los campos de la tabla.



Ejemplo:

idEmp	Nombre	idArea	NroHoras
c0001	Pablo	a0001	45
c0002	Andrea	a0002	30
c0003	Carlos	a0003	32

Buscamos las dependencias funcionales:

idEmp → Nombre

idEmp, idArea → NroHoras

idEmp	Nombre	idEmp	idArea	NroHoras
c0001	Pablo	c0001	a0001	45
c0002	Andrea	c0002	a0002	30
c0003	Carlos	c0003	a0003	32

Buscamos descomponer en diferentes tablas tantas dependencias funcionales consigamos (hay que analizarlas).

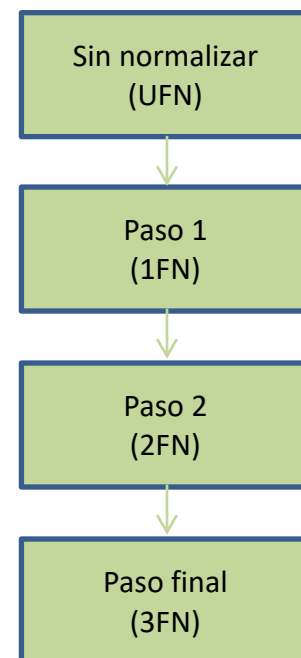
- 3era Forma Normal (3FN o 3NF).

La tabla debe estar en 2FN y a su vez ningún atributo no clave debe estar envuelto en una dependencia transitiva. Debemos chequear las dependencias transitivas de la tabla.

$A \longrightarrow B \longrightarrow C$

A: clave primaria o campos de la tabla.

B, C: el resto de los campos de la tabla.



Ejemplo:

NTorneo	Año	NGanador	DOB
Master 1	1998	Carlos	14 Marzo 1980
Master 2	1998	Andres	12 Diciembre 1979
Master 1	1999	Carlos	21 Enero 1981
Master 3	1999	Susana	13 Febrero 1977
Master 2	1999	Andrea	14 Mayo 1980

Dependencias Funcionales:

$NTorneo, Año \longrightarrow NGanador$

$NGanador \longrightarrow DOB$

Por tanto, DOB depende transitivamente de la clave primaria.

Master 1	2000	Luis	13 Abril 1978	
----------	------	------	---------------	--

NGanador	DOB
Carlos	14 Marzo 1980
Andres	12 Diciembre 1979
Carlos	21 Enero 1981
Susana	13 Febrero 1977
Andrea	14 Mayo 1980
Luis	13 Abril 1978

NTorneo	Año	NGanador
Master 1	1998	Carlos
Master 2	1998	Andres
Master 1	1999	Carlos
Master 3	1999	Susana
Master 2	1999	Andrea
Master 1	2000	Luis

A pesar de poner la tabla en 3FN, notamos que las mismas no se encuentran 1FN, ya que la primera tabla carece de clave primaria y en la segunda puede ocurrir una anomalía de modificación. Por tanto corregimos ambos casos y tenemos:

ID	NGanador	DOB
1	Carlos	14 Marzo 1980
2	Andres	12 Diciembre 1979
3	Carlos	21 Enero 1981
4	Susana	13 Febrero 1977
5	Andrea	14 Mayo 1980
6	Luis	13 Abril 1978

NTorneo	Año	IDfk
Master 1	1998	1
Master 2	1998	2
Master 1	1999	3
Master 3	1999	4
Master 2	1999	5
Master 1	2000	6

La tabla se encuentra en 3FN y podemos pasar a la **fase de implementación** en un manejador de base de datos como Oracle, SQLite o MySQL.