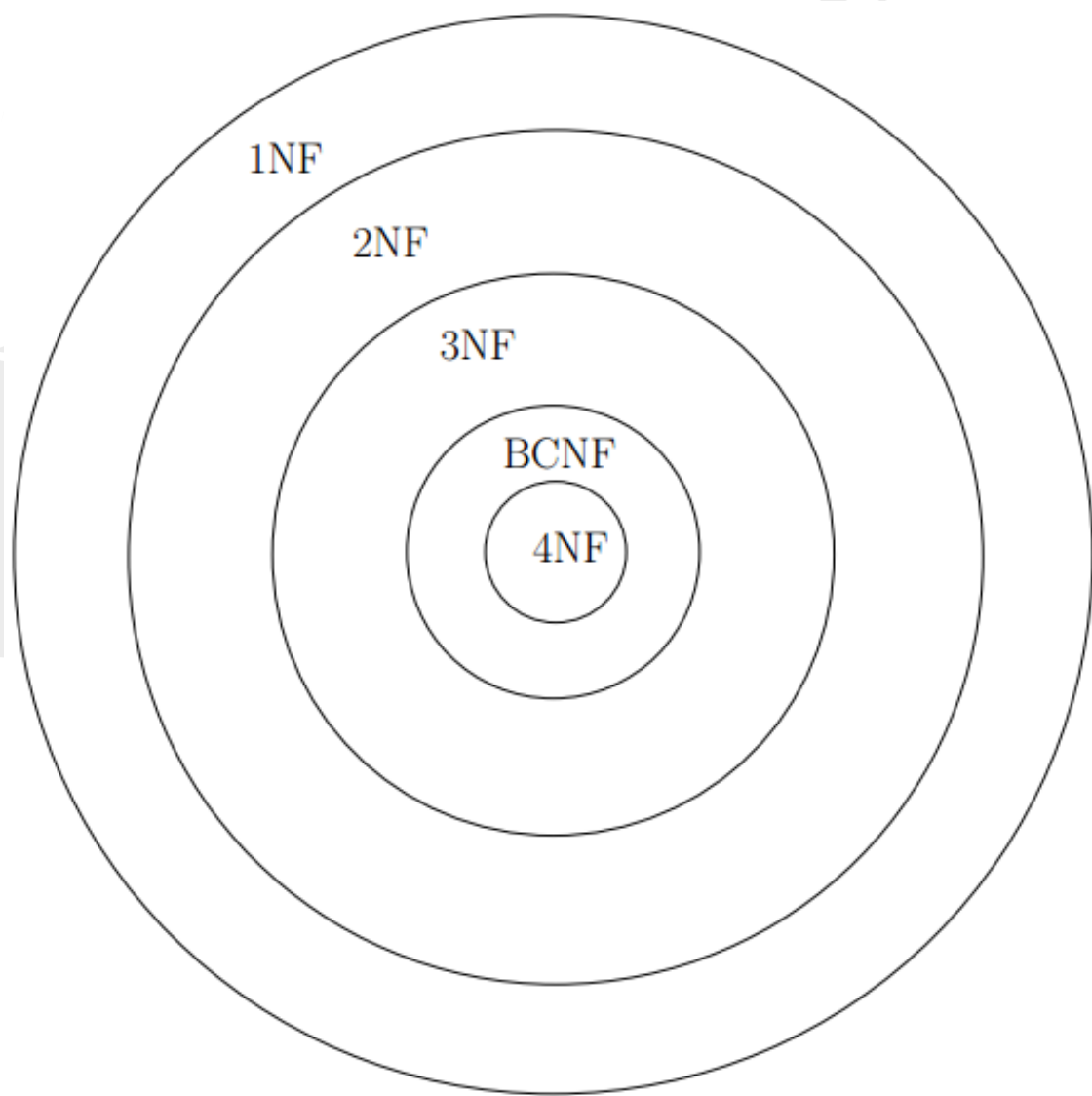


Normalización

Bases de Datos



Primera Forma Normal

- El dominio de todo atributo de cualquier relación debe incluir sólo valores atómicos.
- El valor de cualquier tupla debe ser un elemento simple del dominio.
- No puede ser que para una tupla r y un atributo A el valor de $r[A]$ sea un conjunto de valores atómicos.

Ejemplo

departamento

<i>nom_depto</i>	<i>codigo</i>	<i>fech_creac</i>	<i>telefonos</i>
Informática	A1	01/03/2002	{6354929, 6382276, 2262875}
Marketing	A2	01/01/2002	{6316651, 2775331}
Ventas	A3	01/01/2001	{6382276}
Recursos Humanos	A4	01/01/2003	{2775331}

si $r(R)$ es la relación que viola 1NF, N el atributo que produce el problema y K la clave primaria de $r(R)$.

Solución 1

- Crear una nueva relación $r' (R')$ a partir de remover el atributo N de $r (R)$
- Crear una nueva relación $t (K, N')$ donde K es la clave primaria de $r(R)$ de tal manera que las tuplas cumplan:

$$t[N'] \in r[N] \Leftrightarrow t[K] = r[K]$$

- Esta última relación tiene clave primaria (K, N')
- Eliminar la relación $r (R)$ del esquema de la Base de Datos.

<i>nom_depto</i>	<i>codigo</i>	<i>fech_creac</i>	<i>telefonos</i>
Marketing	A2	01/01/2002	{6316651, 2775331}
Ventas	A3	01/01/2001	{2775331}
Recursos Humanos	A4	01/01/2003	{6382276}



<i>nom_depto</i>	<i>codigo</i>	<i>fech_creac</i>
Informática	A1	01/03/2002
Marketing	A2	01/01/2002
Ventas	A3	01/01/2001
Recursos Humanos	A4	01/01/2003

<i>codigo</i>	<i>telefono</i>
A1	6354929
A1	6382276
A1	2262875
A2	6316651
A2	2775331
A3	6382276
A4	2775331

Solución 2

- Crear una nueva relación $r' (R')$ a partir de remover el atributo N de $r (R)$ y agregar un nuevo atributo N' de tal manera que las tuplas cumplan:

$$r' [N'] \in r [N] \Leftrightarrow r' [K] = r [K]$$

- La clave primaria de $r' (R')$ es (K, N')
- Eliminar la relación $r (R)$ del esquema de la Base de Datos.

<i>nom_depto</i>	<i>codigo</i>	<i>fech_creac</i>	<i>telefonos</i>
Informática	A1	01/03/2002	{6354929, 6382276, 2262875}
Marketing	A2	01/01/2002	{6316651, 2775331}
Ventas	A3	01/01/2001	{2775331}
Recursos Humanos	A4	01/01/2003	{6382276}



<i>nom_depto</i>	<i>codigo</i>	<i>fech_creac</i>	<i>telefono</i>
Informática	A1	01/03/2002	6354929
Informática	A1	01/03/2002	6382276
Informática	A1	01/03/2002	2262875
Marketing	A2	01/01/2002	6316651
Marketing	A2	01/01/2002	2775331
Ventas	A3	01/01/2001	6382276
Recursos Humanos	A4	01/01/2003	2775331

Segunda Forma Normal

Está basada en el concepto de dependencia funcional total:

- sean X , Y y B conjuntos de atributos de cierta relación
- Si $X \rightarrow Y$ y para todo $B \subsetneq X$ se cumple que $B \not\rightarrow Y$, entonces se dice que Y es totalmente funcionalmente dependiente de X (para definir Y se necesita X completo).
- Si no, se dice que Y es parcialmente funcionalmente dependiente de X

* Sea A un subconjunto de B tal que $A \neq B$. Entonces se dice que A es un **subconjunto propio** de B , y se denota por $A \subsetneq B$.

Formulación-2NF

Para entender la idea de 2NF veremos inicialmente un caso particular simple en donde queda claro su aplicación.

- Supondremos que todas las relaciones tienen **una única llave candidata** lo que quiere decir que las llaves primarias se encuentran definidas.
- Un esquema de Base de Datos con la anterior característica se encuentra en 2NF si está en 1NF y toda relación $r(R)$ con llave primaria K cumple que para cualquier atributo A que no sea parte de la llave primaria, A es totalmente funcionalmente dependiente de K .

$K \rightarrow A$ es una dependencia funcional total

- si la llave primaria de una relación está compuesta por sólo un atributo, la propiedad se cumple siempre.

<i>rut_emp</i>	<i>cod_proy</i>	<i>horas</i>	<i>nom_emp</i>	<i>nom_proy</i>	<i>lugar_proy</i>
----------------	-----------------	--------------	----------------	-----------------	-------------------

Claramente la única clave candidata es (rut, cod_proy) y podemos decir que:

- $rut, cod_proy \rightarrow horas$ es una dependencia funcional total.
- $rut, cod_proy \rightarrow nom_emp$ es una dependencia funcional parcial ya que se cumple $rut \rightarrow nom_emp$.
- $rut, cod_proy \rightarrow nom_proy, lugar_proy$ es una dependencia funcional parcial ya que $cod_proy \rightarrow nom_proy, lugar_proy$.

No se encuentra en 2NF ya que *nom_emp*, *nom_proy* y *lugar_proy* dependen funcionalmente en forma parcial de la llave primaria.

Solución

Primero convertir en 1NF. Sea $r(R)$ una relación que viola 2NF, y K la llave primaria de $r(R)$

- Identificar un conjunto de atributos A que depende funcionalmente en forma parcial de K , y el subconjunto $K' \subset K$ del que depende funcionalmente en forma total.
- Crear una nueva relación $r'(R')$ que resulta de $r(R)$ al eliminar el conjunto de atributos A .
- Crear una nueva relación $t(K', A)$ de tal manera que las tuplas cumplan:
$$t[A] = r[A] \Leftrightarrow t[K'] = r[K']$$
- La llave primaria de $t(K', A)$ es K' .
- Eliminar la relación $r(R)$ del esquema.

Solución

A pesar de que $\pi_t(K', A)$ está en 2NF, $\pi_{r'}(R')$ posiblemente aún no se encuentra en 2NF, por lo que se debe repetir el proceso hasta que el esquema esté en 2NF.

<u><i>rut_emp</i></u>	<u><i>cod_proy</i></u>	<i>horas</i>	<i>nom_emp</i>	<i>nom_proy</i>	<i>lugar_proy</i>
-----------------------	------------------------	--------------	----------------	-----------------	-------------------



<u><i>rut_emp</i></u>	<u><i>cod_proy</i></u>	<i>horas</i>	<i>nom_proy</i>	<i>lugar_proy</i>
-----------------------	------------------------	--------------	-----------------	-------------------

<u><i>rut_emp</i></u>	<u><i>nom_emp</i></u>
-----------------------	-----------------------



<u><i>rut_emp</i></u>	<u><i>cod_proy</i></u>	<i>horas</i>
-----------------------	------------------------	--------------

<u><i>rut_emp</i></u>	<u><i>nom_emp</i></u>
-----------------------	-----------------------

<u><i>cod_proy</i></u>	<i>nom_proy</i>	<i>lugar_proy</i>
------------------------	-----------------	-------------------

Ejemplo 2

<u><i>cod_propiedad</i></u>	<i>municipio</i>	<i>numero_lote</i>	<i>area</i>	<i>precio</i>	<i>impuesto</i>
-----------------------------	------------------	--------------------	-------------	---------------	-----------------

Claves candidatas:

- *cod propiedad*
- *municipio, numero lote*

Más la siguiente dependencia funcional:

- *municipio* → *impuesto*

<u><i>cod_propiedad</i></u>	<i>municipio</i>	<i>numero_lote</i>	<i>area</i>	<i>precio</i>	<i>impuesto</i>
-----------------------------	------------------	--------------------	-------------	---------------	-----------------



<u><i>cod_propiedad</i></u>	<i>municipio</i>	<i>numero_lote</i>	<i>area</i>	<i>precio</i>
-----------------------------	------------------	--------------------	-------------	---------------

<u><i>municipio</i></u>	<i>impuesto</i>
-------------------------	-----------------

Ejercicios

Normalizar hasta 2FN

R: A, B, C, D, E.

- A \rightarrow B
- A \rightarrow D
- C \rightarrow E

Ejercicios

Normalizar hasta 2FN

R: A, B, C, D, E, F, G.

- F \rightarrow B
- E \rightarrow D
- A \rightarrow C
- G \rightarrow E
- B \rightarrow F

Fin

Por ahora...

Tercera Forma Normal

Está basada en el concepto de dependencia funcional transitiva:

- sean X, Y y Z conjuntos de atributos de cierta relación.
- Si $X \rightarrow Y$ y existe un conjunto $Z \sim X$, tal que $X \rightarrow Z$ y $Z \rightarrow Y$ se dice que Y depende funcionalmente en forma transitiva de X a través de Z .

Ejemplo

<u>nom_emp</u>	suelo	cod_dept	fecha_ing	nom_dept	fecha_crear
----------------	-------	----------	-----------	----------	-------------

Supongamos que tenemos el siguiente esquema de relación empleado-departamento.

- nom_emp llave candidata única (llave primaria)
- cod_dept → nom_dept, fecha_crear

En el ejemplo nom_dept y fecha_crear dependen funcionalmente en forma transitiva de nom_emp

nom_emp → cod_dept y cod_dept → nom_dept, fecha_crear.

3NF – Formulación

- Supondremos que todas las relaciones tienen una única llave candidata lo que quiere decir que las llaves primarias se encuentran definidas (cómo lo hicimos con 2NF).
- Un esquema de Base de Datos se encuentra en 3NF si está en 2NF y toda relación $r(R)$ con llave primaria K cumple que para cualquier atributo A que no sea parte de la llave primaria en R , A no depende funcionalmente en forma transitiva de K

$$\nexists B \text{ tal que } K \rightarrow B \text{ y } B \rightarrow A$$

¿3NF?

<u>nom_emp</u>	suelo	cod_dept	fecha_ing	nom_dept	fecha_creat
----------------	-------	----------	-----------	----------	-------------

3NF – Solución

- Primero convertir en 2NF. Sea $r(R)$ la relación que viola 3NF, y K la clave primaria de $r(R)$
- Identificar un conjunto de atributos A ($A \not\subseteq K$) que depende funcionalmente en forma transitiva de K , y el conjunto B que define la transitividad
- Crear una nueva relación $r'(R')$ que resulta de $r(R)$ al eliminar el conjunto de atributos A

3NF – Solución

- Crear una nueva relación $t(B, A)$ tal que:

$$t[A] = r[A] \Leftrightarrow t[B] = r[B]$$

- La clave primaria de $t(B, A)$ es B , se debe establecer B como llave foránea en $r'(R')$.
- Eliminar la relación $r(R)$ del esquema.
- $r'(R')$ posiblemente aún no se encuentra en 3NF, por lo que se debe repetir el proceso hasta que el esquema esté en 3NF.

3NF – Solución (For Dummies)

<u>nom_emp</u>	suelo	cod_dept	fecha_ing	nom_dept	feh_creac
----------------	-------	----------	-----------	----------	-----------



<u>nom_emp</u>	suelo	cod_dept	fecha_ing
----------------	-------	----------	-----------

<u>cod_dept</u>	nom_dept	feh_creac
-----------------	----------	-----------

3NF – Formulación (cont.)

- En la anterior formulación supusimos que las relaciones tenían una única llave candidata y por lo tanto la llave primaria estaba fija.
- En general este supuesto puede no ser cierto y entonces 3NF cambian en parte su formulación:
 - Un esquema está en 3NF si para cada relación $r(R)$, si un atributo A no es parte de una llave candidata en R , entonces A no depende en forma transitiva de ninguna llave candidata de R .
 - Es simplemente una generalización del caso anterior pero ahora tomando en cuenta todas las llaves candidatas (no sólo la primaria).

3NF – Formulación (cont.)

Para cada dependencia funcional no trivial $X \rightarrow A$ (o sea tal que $A \not\subseteq X$) que existe en $r(R)$ se cumple que:

- X es una clave candidata de R .
- A es parte de una clave candidata de R .

La solución es similar a la anterior salvo que se toman en cuenta en total las llaves candidatas, no sólo las llaves primarias.

3NF – Ejemplo

<u><i>cod_propiedad</i></u>	<i>municipio</i>	<i>numero_lote</i>	<i>area</i>	<i>precio</i>
<u><i>municipio</i></u>		<i>impuesto</i>		

(municipio, número lote) es también llave candidata y tenemos la dependencia funcional:

municipio → impuesto.

Agregamos la dependencia funcional

- área → precio

3NF – Ejemplo

- No se encuentra en 3NF ya que precio depende en forma transitiva de una llave candidata:

`cod_propiedad → area → precio.`

- Equivalentemente, `area → precio` es una dependencia funcional no trivial, `area` no es una llave candidata y `precio` no es parte de ninguna llave candidata, luego el esquema no se encuentra en 3NF.

Solución

<u><i>cod_propiedad</i></u>	<i>municipio</i>	<i>numero_lote</i>	<i>area</i>
-----------------------------	------------------	--------------------	-------------

<u><i>municipio</i></u>	<i>impuesto</i>	<u><i>area</i></u>	<i>precio</i>
-------------------------	-----------------	--------------------	---------------

BCNF – Forma Normal de Boyce-Codd

- Propuesta inicialmente como una forma simplificada de 3NF.
- Es muy simple de formular, pero resultó ser más estricta que 3NF.
- Si un esquema está en BCNF entonces está en 3NF, pero no necesariamente al revés.
- Un esquema está en BCNF si cada una de sus relaciones $r(R)$ cumple:
 - Para toda dependencia funcional no trivial $X \rightarrow Y$ en R , X es una llave candidata.

BCNF – Forma Normal de Boyce-Codd

- La única diferencia es que ahora los atributos que son parte de una llave candidata sólo pueden depender de otras llaves candidatas.

<u><i>cod_propiedad</i></u>	<i>municipio</i>	<i>numero_lote</i>	<i>area</i>
<u><i>municipio</i></u>	<i>impuesto</i>	<u><i>area</i></u>	<u><i>precio</i></u>

restricciones:

- Supongamos que tenemos muchos lotes pero sólo los municipios de Curicó y Talca.
- Las superficies de los lotes de Curicó son 0.5, 0.6, 0.7, 0.8 y 0.9, y de los lotes de Talca son 1.0, 1.1, 1.2, 1.3.
- Obtenemos la nueva dependencia $area \rightarrow municipio$.

BCNF – Forma Normal de Boyce-Codd

- El esquema sigue estando en 3NF, ya que municipio es parte de una llave candidata (no existen dependencias transitivas de atributos no parte de llaves candidatas).
- Sin embargo el esquema no está en BCNF ya que el atributo area no es una llave candidata.

BCNF – Forma Normal de Boyce-Codd

<u><i>cod_propiedad</i></u>	<i>municipio</i>	<i>numero_lote</i>	<i>area</i>
-----------------------------	------------------	--------------------	-------------

<u><i>municipio</i></u>	<i>impuesto</i>
-------------------------	-----------------

<u><i>area</i></u>	<i>precio</i>
--------------------	---------------



<u><i>cod_propiedad</i></u>	<i>numero_lote</i>	<i>area</i>
-----------------------------	--------------------	-------------

<u><i>area</i></u>	<i>municipio</i>
--------------------	------------------

<u><i>municipio</i></u>	<i>impuesto</i>
-------------------------	-----------------

<u><i>area</i></u>	<i>precio</i>
--------------------	---------------

BCNF – Forma Normal de Boyce-Codd

- En general la forma de llevar un esquema a uno equivalente en 3NF es bastante simple, informalmente:
 - Identificar la relación R y la dependencia funcional $X \rightarrow A$ tal que X no es llave en R .
 - Crear una nueva relación con atributos X y A , tomando X como llave primaria.
 - Dejar X en R como llave foránea (eliminar los atributos correspondientes de R).

Normalización – Comparaciones

- 1NF \Rightarrow atributos atómicos.
- 2NF \Rightarrow atributos no parte de llaves candidatas dependen en forma total de todas las llaves candidatas.
- 3NF \Rightarrow atributos no parte de llaves candidatas no dependen en forma transitiva de ninguna llave candidata.
- BCNF \Rightarrow todas las dependencias funcionales son desde llaves (si $X \rightarrow Y$ entonces X es llave).

Ejercicios

Normalizar hasta BCNF

1. R: A, B, C, D, E.

- A \twoheadrightarrow B
- A \twoheadrightarrow D
- C \twoheadrightarrow E

2. R: A, B, C, D, E, F, G.

- F \twoheadrightarrow B
- E \twoheadrightarrow D
- A \twoheadrightarrow C
- G \twoheadrightarrow E
- B \twoheadrightarrow F

Ejercicios

Normalizar hasta BCNF

3. R: A, B, C, D, E, F, G.

- C \rightarrow E
- A \rightarrow C
- D \rightarrow B, F
- B \rightarrow D, G

4. R: A, B, C, D, E, F

- BD \rightarrow E
- CD \rightarrow A
- E \rightarrow C
- B \rightarrow D

Fin

Por ahora...