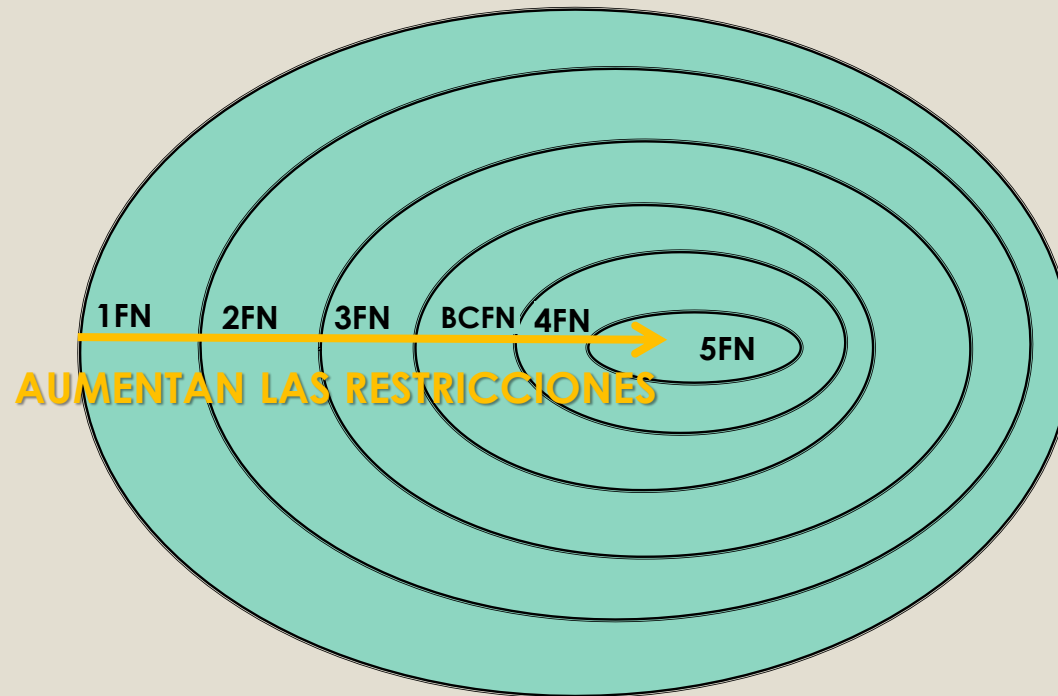




Unidad 3: Normalización

Diseño lógico de una
base de datos

Normalización (Formas Normales)



Primera Forma Normal



Una relación R está en 1FN si los dominios de todos sus atributos son atómicos



Toda relación R está en 1FN, por definición

Segunda Forma Normal

Basada en el concepto de dependencia **funcional completa/total**:

- ✓ Una dependencia funcional $X \rightarrow Y$ es **completa** si la eliminación de cualquier atributo A de X hace que la dependencia no sea válida. Es decir, **$X-A$** no determina funcionalmente a Y .
- ✓ Una dependencia funcional $X \rightarrow Y$ es **parcial** si es posible eliminar algún atributo A de X y la dependencia sigue siendo válida.

Ejercicio, analicemos:

NroPat + Modelo \rightarrow Marca ?

Segunda Forma Normal

Una relación **R está en 2FN** si todo atributo no primo A de R depende funcionalmente en forma completa de las claves candidatas de R.

Analicemos el siguiente ejemplo:

ventas = (nroFac, nroProd, nombreProd, rubro, precioActual, cantVen, precioVta, fechaFac)

1. ¿Dependencias Funcionales?
2. ¿Claves candidatas?



Segunda Forma Normal

ventas = {nroFac, nroProd, nombreProd, rubro, precioActual, cantVen, precioVta, fechaFac}

Dependencias Funcionales

DF1: nroFac -> fechaFac

DF2: nroProd -> nombreProd, rubro, precioActual

DF3: nroFac+nroProd -> cantVen, precioVta

Claves candidatas

CC = nroFac+nroProd

Recordando... **R está en 2FN** si todo atributo no primo A de R depende funcionalmente en forma completa de las claves candidatas de R

Atributos no primos = fechaFac, nombreProd, rubro, precioActual, cantVen, precioVta

La relación **compra no está en 2FN** puesto que en DF1 y DF2 se verifica que existen atributos no primos que dependen en forma parcial de la clave

Segunda Forma Normal

Solución:

Si una relación no está en 2FN, debemos descomponerla y crear una nueva relación para cada clave parcial con su/s atributo/s dependiente/s (manteniendo las dependencias).

Ventas = (nroFac, nroProd, nombreProd, rubro, precioActual, cantVen, precioVta, fechaFac)

Ventas1 = {nroFac, fechaFac}

Ventas2 = {nroProd, nombreProd, rubro, precioActual}

Ventas3 = { nroFac, nroProd, cantVen, precioVta}

Dependencias Funcionales

DF1: nroFac -> fechaFac

DF2: nroProd -> nombreProd, rubro, precioActual

DF3: nroFac+nroProd -> cantVen, precioVta

NOTA: El join (unión) de las 3 relaciones debe generar la relación original.

Descomposiciones válidas

Toda descomposición debe cumplir una restricción
(no sólo para la 2FN, sino para todas):

- **No generar pérdida de información**, es decir, el join (lo veremos mas adelante) de las proyecciones genera la relación original.
Inclusive, provocan ganancia de información.
- Permiten registrar información que en la relación original era imposible. (*Recordar lo explicado en problemas de actualización*)

Tercer Forma Normal

- ✓ Una relación R está en 3FN si:
 - ✓ Está en 2FN
 - ✓ **No existen dependencias funcionales** entre los **atributos no primos de R**, es decir, los **atributos no primos son mutuamente independientes**.



Tercer Forma Normal

Analicemos el siguiente ejemplo:

producto = (nroProd, nombreProd, rubro, cuitProveedor *, direProveedor)



****Nota:** Cada producto es provisto solamente por un proveedor*

- Claves candidatas

cc = {nroProd}

- Dependencias Funcionales

DF1: cuitProveedor -> direProveedor

DF2: nroProd -> nombreProd, rubro, cuitProveedor

La relación **producto no está en 3FN**, puesto DF1 muestra una dependencia entre los atributos no primos cuitProveedor y direProveedor, es decir, no son mutuamente independientes

Tercera Forma Normal

Solución:

Si una relación no está en 3FN, descomponerla creando otra relación que **contenga el/los atributo/s no claves que determinan funcionalmente al otro/s atributo/s no clave.**

Nota: “Tener cuidado con las posibles descomposiciones”

Tercera Forma Normal

producto = {nroProd, nombreProd, rubro, cuitProveedor, direProveedor}

Dependencias Funcionales y claves

DF1: cuitProveedor -> direProveedor

DF2: nroProd -> nombreProd, rubro, cuitProveedor

cc = nroProd

Solución1:

Producto1 = {nroProd, nombreProd, rubro, cuitProveedor}

Producto2 = {cuitProveedor, direProveedor}

Solución2 :

Producto1 = {nroProd, nombreProd, rubro, cuitProveedor}

Producto2 = {nroProd, direProveedor}

Analicemos las 2 alternativas, ¿cuál será mejor? ¿ por qué?

¿Dudas?



N
O
R
M
A
L
I
Z
A
C
I
Ó
N

Forma Normal de Boyce y Codd

Una relación R está en FNBC si todo determinante es clave candidata

IMPORTANTE:

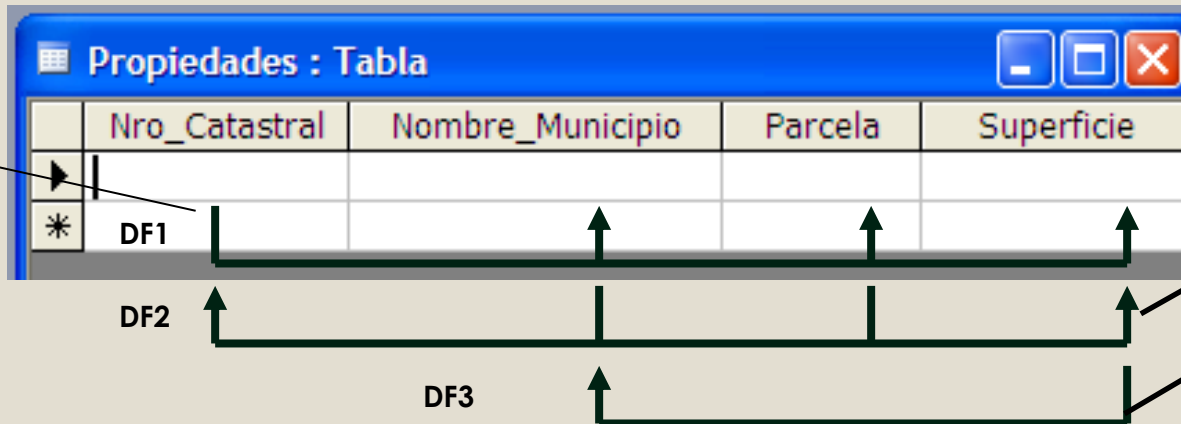
- No refiere a una FN anterior



Forma Normal de Boyce y Codd

Veamos un ejemplo:

Cada parcela es identificada por su nro. catastral



Cada parcela es identificada dentro de un municipio, por su nro. de parcela

Las sup. de las parcelas de cada municipio son diferentes, pero, no existe una sup. que corresponda a más de un municipio.

Determinantes:

- Nro_Catastral
- Nombre_Municipio+Parcela

• Superficie

No es clave candidata

Claves candidatas:

- Nro_Catastral
- Nombre_Municipio+Parcela

FNBC

Forma Normal de Boyce y Codd

Al descomponer:

	Nro_Catastral	Superficie	Parcela
▶			

A diagram below the table shows a horizontal line starting from the Nro_Catastral column, with two vertical arrows pointing up to the Superficie and Parcela columns, indicating that Nro_Catastral determines both Superficie and Parcela.

Determinantes: Nro_Catastral

Claves candidatas: Nro_Catastral



BCNF

	Superficie	Nombre_Municipio
▶		

A diagram below the table shows a horizontal line starting from the Superficie column, with a vertical arrow pointing up to the Nombre_Municipio column, indicating that Superficie determines Nombre_Municipio.

Determinantes: Superficie

Claves candidatas: Superficie



BCNF

Terminamos!



N
O
R
M
A
L
I
Z
A
C
I
Ó
N