

# Normalización

## Normalización

Es el proceso mediante el cual se va reduciendo un conjunto de relaciones dado, a una forma más deseable. En este contexto “deseable” significa más sencilla, más simple. Otra manera de definirla es “ir descomponiendo relaciones en otras relaciones mejores en términos de las operaciones que se ejecutan sobre ellas” [1].

Existen varias formas normales para las relaciones:

1NF,

*serán estudiadas*

2NF,

3NF,

BCNF,

4NF y 5NF (PJNF)

## Dependencias Funcionales - Restricciones

Se dice que una relación está en una determinada forma normal si satisface un cierto conjunto de restricciones.

### **Noción de Dependencia Funcional**

Dada una relación R, el atributo Y de R depende funcionalmente del atributo X de R si para una tupla dada el valor en X de R determina el valor en Y de R, es decir, si:

$$R(X) \rightarrow R(Y)$$

Supongamos que tenemos un conjunto de relaciones:

Pieza(p#, nombre-p, color-p, peso-p, ciudad-alm)

Vendedor(v#, nombre-v, estatus, ciudad)

Venta(v#, p#, cantidad)

En estas relaciones se dan ciertas dependencias funcionales, tales como:

Pieza(p#)  $\rightarrow$  Pieza(nombre-p)

Pieza(p#)  $\rightarrow$  Pieza(color-p)

Pieza(p#)  $\rightarrow$  Pieza(peso-p)

Pieza(p#)  $\rightarrow$  Pieza(ciudad-alm)

Vendedor(v#)  $\rightarrow$  Vendedor(nombre-v)

Vendedor(v#)  $\rightarrow$  Vendedor(estatus)

Vendedor(v#)  $\rightarrow$  Vendedor(ciudad)

Venta(s#,p#)  $\rightarrow$  Venta(cantidad)

No necesariamente, la dependencia funcional de Y respecto de X viene dada porque el hecho de que X es una clave candidata. De manera que una definición que abarque ese punto se puede escribir como:

## Definición de Dependencia Funcional

“Dada una relación R, el atributo Y de R depende funcionalmente del atributo X de R si y siempre que dos tuplas de R concuerden en su valor de X, deben por fuerza, concordar en su valor de Y” [1].

### EJEMPLO

Supongamos que existiese una relación

$Venta'(v\#, p\#, cantidad, ciudad)$

en la relación  $Venta'$   $v\#$  no es clave candidata, sin embargo,

$Venta'(v\#) \rightarrow Venta'(ciudad)$

Por esa misma razón  $ciudad$  no depende completamente de la clave primaria  $(v\#, p\#)$  .

**Se dice, entonces que  $ciudad$  tiene dependencia funcional no completa de la clave primaria.**

### Definición de Dependencia Funcional Completa

“ Se dice que  $Y$  de  $R$  depende funcionalmente por completo de  $X$  de  $R$ , si depende funcionalmente de  $X$  y no depende funcionalmente de ningún subconjunto de  $X$ ” [1].

Dicho en otras palabras, no existe un subconjunto  $Z$  de atributos componentes de  $X$  de los cuales  $Y$  dependa funcionalmente.

## Definición de la Primera Forma Normal (1NF)

“Una relación está en 1NF si todos los dominios simples subyacentes contienen sólo valores atómicos”

Relación No Normalizada

V#	Pedido	
	p#	cantidad
01	p1	100
	p2	150
	p3	200

Relación Normalizada

v#	p#	cantidad
01	p1	100
01	p2	150
02	p3	200



## Definición de la Segunda Forma Normal (2NF)

“Una relación está en 2NF sii está en 1NF y todos los atributos no clave (o que no formen parte de la clave) dependen por completo de la clave primaria”.

M

<u>v#</u>	estatus	ciudad	<u>p#</u>	cantidad
01	10	ccs	p1	100
01	10	ccs	p2	150
02	20	vcia	p3	200
03	30	b/to	p4	50
03	30	b/to	p5	100
04	10	ccs	p4	50

$M(\text{estatus}) \rightarrow M(\text{ciudad}), M(\text{ciudad}) \rightarrow M(\text{estatus})$

$M(v\#) \rightarrow M(\text{estatus})$

$M(v\#) \rightarrow M(\text{ciudad})$

es decir, no dependen por completo  
de la clave primaria  $(v\#, p\#)$

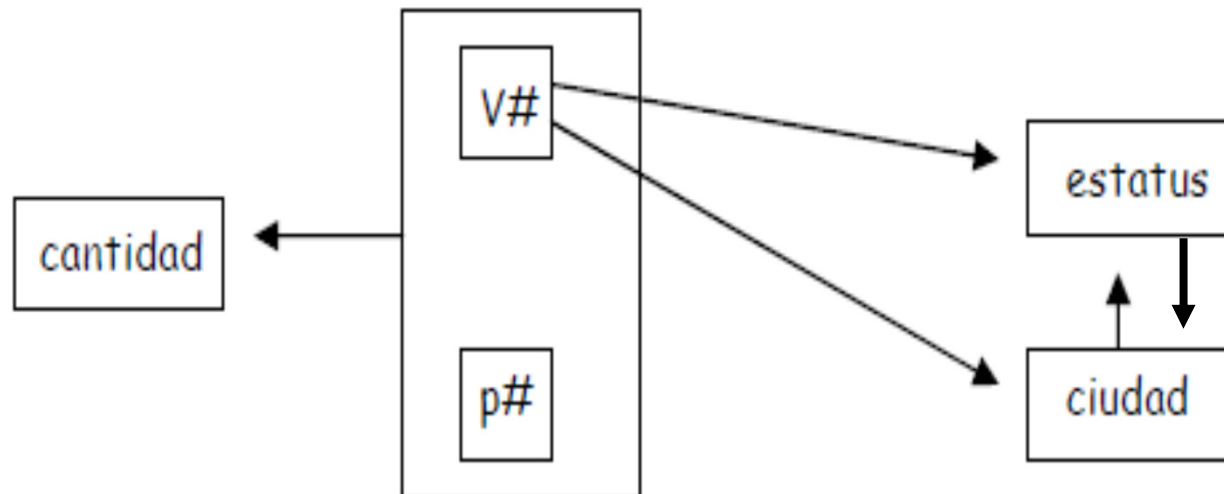
$M(\text{estatus}) \rightarrow M(\text{ciudad})$ ,  $M(\text{ciudad}) \rightarrow M(\text{estatus})$

$M(v\#) \rightarrow M(\text{estatus})$

$M(v\#) \rightarrow M(\text{ciudad})$

es decir, no dependen por completo  
de la clave primaria  $(v\#, p\#)$

## Diagrama de Dependencias Funcionales



## Problemas que presenta la Relación M

M

<u>V#</u>	estatus	ciudad	<u>p#</u>	cantidad
01	10	ccs	p1	100
01	10	ccs	p2	150
02	20	vcia	p3	200
03	30	b/to	p4	50
03	30	b/to	p5	100
04	10	ccs	p4	50

Hay redundancias:

- 1) Todas las tuplas donde aparece “ccs” tiene el estatus 10. Si hubieran 1000 tuplas con “ccs” se repetiría 1000 que su estatus es 10.
- 2) De igual forma, todas las tuplas donde aparece el vendedor también aparece la ciudad a donde está asignado.

## Problemas que presenta la Relación M

M

<u>v#</u>	estatus	ciudad	<u>p#</u>	cantidad
01	10	ccs	p1	100
01	10	ccs	p2	150
02	20	vcia	p3	200
03	30	b/to	p4	50
03	30	b/to	p5	100
04	10	ccs	p4	50

### Anomalías de Actualización:

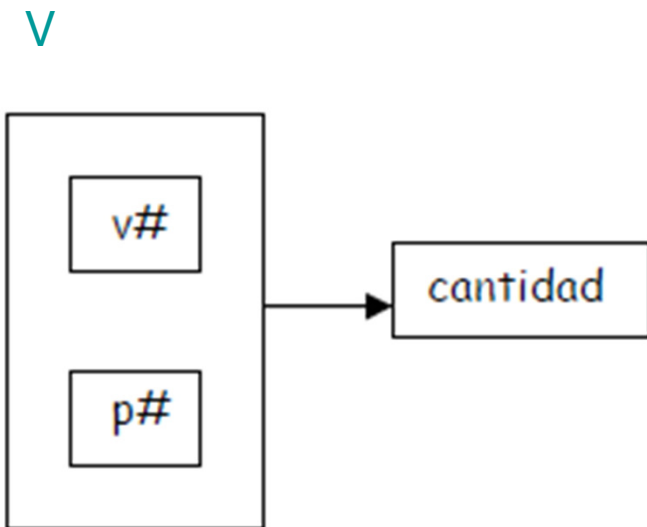
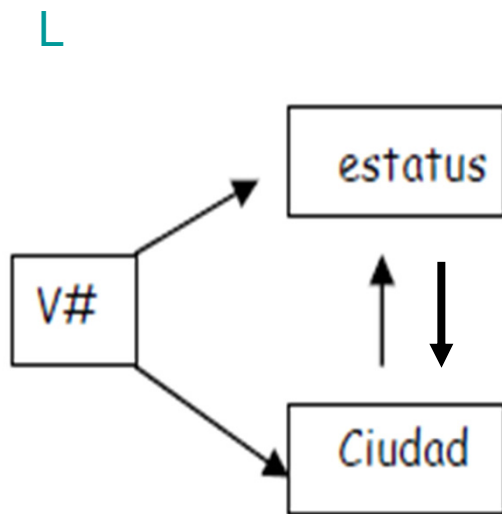
- 1) Inserción:** si un vendedor v# no ha vendido piezas, aunque sepamos cuál es la ciudad en donde está asignado y su estatus, **no se puede ingresar a la base de datos.**
- 2) Eliminar:** si sólo hay una tupla para un v# con la venta de un vendedor y se elimina la venta, también **se pierde la información** de cuál es la ciudad y el estatus que ese vendedor tiene.
- 3) Actualización:** si v# se cambia de ciudad tenemos que **buscar en todas las tuplas** donde aparece para actualizar su estatus y ciudad.

# SOLUCIÓN

Crear dos relaciones donde no ocurran estas anomalías:

$L(\underline{v\#}, \text{estatus}, \text{ciudad})$

$V(\underline{v\#}, \underline{p\#}, \text{cantidad})$



¿Se solucionan las anomalías de actualización?

En general, el problema real de la relación M era que se estaba mezclando información de diferente tipo.

M

<u>V#</u>	estatus	ciudad	<u>p#</u>	cantidad
01	10	ccs	p1	100
01	10	ccs	p2	150
02	20	vcia	p3	200
03	30	b/to	p4	50
03	30	b/to	p5	100
04	10	ccs	p4	50

## Definición de Dependencia Funcional Transitiva

Sea la relación R donde X,Y y Z son un subconjunto de atributos de R, se dice que Z es transitivamente dependiente de X si existe Y tal que:

$$R(X) \rightarrow R(Y) \quad \text{y} \quad R(Y) \rightarrow R(Z) \quad \text{entonces} \quad R(X) \rightarrow R(Z)$$

### Recordar que:

#### *Atributos No Claves:*

Cualquier atributo que no es clave o que no forma parte de la clave.

#### *Atributos Independientes:*

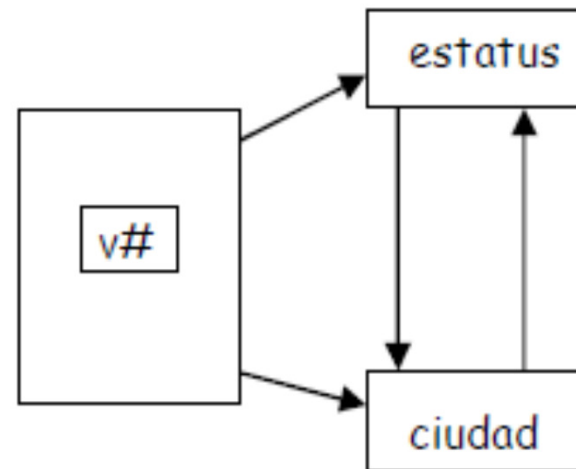
Son atributos que no dependen funcionalmente de otros atributos.

Por ejemplo, en la relación Pieza, color-p y peso-p son atributos independientes entre sí.

## Definición de la Tercera Forma Normal (3NF)

“Una relación R está en 3NF si está en 2NF y todos los atributos no clave dependen de manera no transitiva de la clave primaria.”

Ejemplo: L(v#, estatus, ciudad)





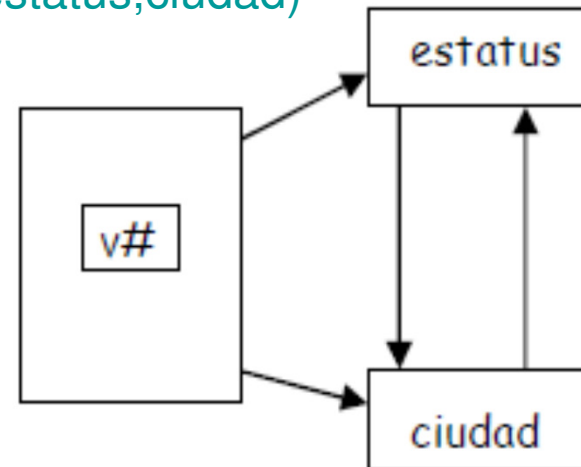
## Problemas de L debido a la dependencia transitiva

a.-**Insertar**: no se puede insertar el *estatus* de una ciudad mientras no halla un vendedor en esa ciudad.

b.-**Eliminar**: si se elimina la única tupla que contenga información del *estatus* de una ciudad, se pierde esa información y no se puede volver a cargar como se dijo en "a".

c.-**Actualizar**: si se efectúa un cambio de *estatus* para una ciudad dada hay que actualizar varias veces según hallan vendedores asociados a esa ciudad.

$L(\underline{v\#}, \text{estatus}, \text{ciudad})$



### SOLUCIÓN

Crear dos relaciones donde no ocurran estas anomalías:

VC(v#,ciudad)  
CS(ciudad,estatus)

ó

VC(v#,estatus)  
CS(ciudad,estatus)

¿Se solucionan las anomalías de actualización?

Ambas relaciones están en 3NF.

## Definición de la Forma Normal Boyce-Codd(BCNF)

“Una relación R está en la BCNF sii todo determinante es una clave candidata en R.”

### Definición de Determinante:

Un atributo o conjunto de atributos es un determinante si determina *por completo* a otro atributo.

## EJEMPLO

Si se tiene la relación:

$SSP(\underline{v\#}, \underline{p\#}, ci-v, cantidad)$

Dado que  $ci-v$  es única para cada vendedor entonces se tienen las siguientes claves candidatas (o alternas) en la relación  $SSP$ :

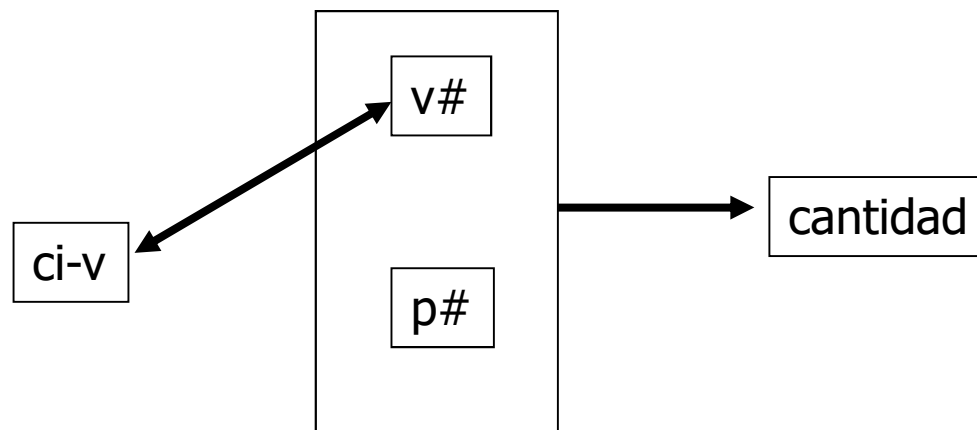
Claves candidatas =  $\{ (\underline{v\#}, \underline{p\#}), (ci-v, \underline{p\#}) \}$

## EJEMPLO

$SSP(\underline{v\#}, \underline{p\#}, ci-v, cantidad)$

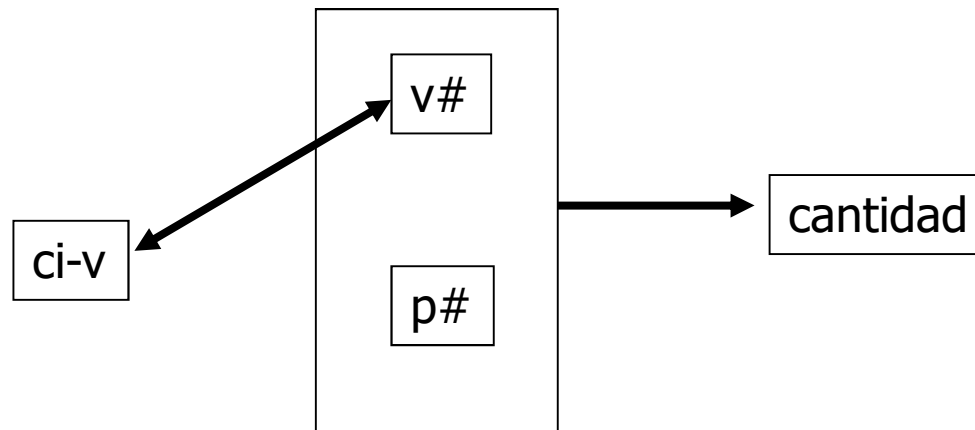
Claves candidatas =  $\{ (v\#, p\#), (ci-v, p\#) \}$

El diagrama de dependencias funcionales de SSP es:



## SSP(v#,p#,ci-v,cantidad)

SSP en 3NF, sin embargo, contiene gran cantidad de redundancia y, por ende, tiene problemas potenciales para la actualización. Además, que no se puede guardar la información de la ci-v del vendedor hasta que él vende algo. Esto también acarrea problemas en la eliminación.



## SOLUCIÓN

Crear dos relaciones:

$SS(\underline{ci-v}, \underline{v\#})$  o  $SS(\underline{ci-v}, v\#)$  y  
 $SP(\underline{ci-v}, \underline{p\#}, cantidad)$

O bien,

$SS(ci-v, \underline{v\#})$  o  $SS(\underline{ci-v}, v\#)$  y  
 $SP(\underline{v\#}, \underline{p\#}, cantidad)$

Ambas combinaciones están en BCNF.

## Fuentes consultadas:

**[1] Silberchatz, Korth. ,**  
"Fundamentos de Bases de Datos".

**[2] Prof. Elsa Liliana Tovar.**  
Notas de clase compiladas entre 1997-2016.