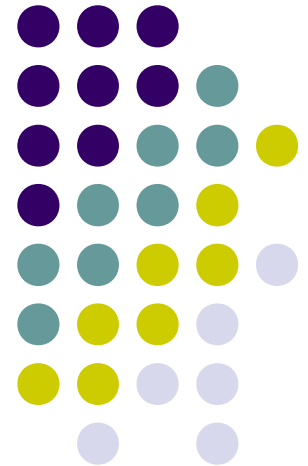
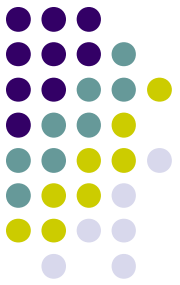


Diseño de Bases de Datos

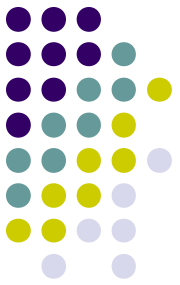
Normalización





Contenidos

- Problemas del mal diseño
- Dependencia Funcionales
- Descomposición / Propiedades Deseables
- Conservación de la Información
- Dependencias Funcionales / Conservación
- Formas Normales: 1FN, 2FN, 3FN, **FNBC**
- Dependencias Multivaluadas
- 4 Forma Normal



Un ejemplo

Proveedores (cod-proveedor, nom-proveedor, cod-insumo, precio)
se divide en:

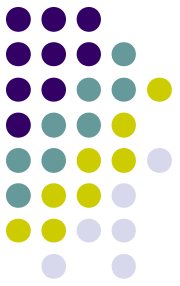
DatosProveedor (cod-proveedor, nom-proveedor)

Suministros (cod-proveedor, cod-insumo, precio)

Para la consulta: “*nombre del proveedor que vende el insumo de código 103*”

- Hacer el join natural entre **DatosProveedor** y **Suministros**

Sin embargo ...



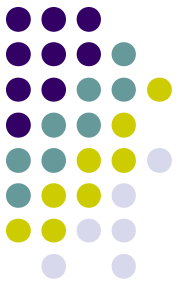
Una división mal hecha puede traer problemas:

IP (cod-insumo, precio)

NDP (cod-proveedor, nom-proveedor, precio)

Pierde información

Ejemplo (1)

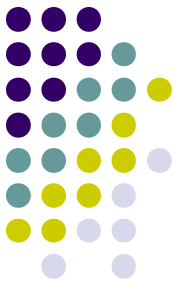


codprov	nomprow	codinsumo	precio
P1	Silva	100	200
P1	Silva	103	70
P2	Morales	201	200
P3	Gallardo	305	100
P3	Gallardo	390	70

PI	
codinsumo	precio
100	200
103	70
201	200
305	100
390	70

NDP		
codprov	nomprow	precio
P1	Silva	200
P1	Silva	70
P2	Morales	200
P3	Gallardo	100
P3	Gallardo	70

Ejemplo (2)



El join natural
entre IP y NDP

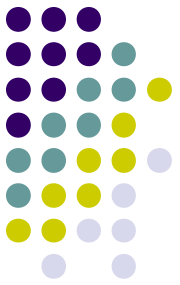
codprov	nomprov	codinsumo	precio
P1	Silva	100	200
P1	Silva	201	200
P1	Silva	103	70
P1	Silva	390	70
P2	Morales	100	200
P2	Morales	201	200
P3	Gallardo	305	100
P3	Gallardo	103	70
P3	Gallardo	390	70

*“nombre de los proveedor
que vende el insumo de
código 103”*

P1	Silva	103	70
P3	Gallardo	103	70

**Hay más información
que en la tabla
original**

Otro ejemplo: Proyección



$\Pi_{\text{placa, marca}}(\text{Auto})$

$\Pi_{\text{marca, modelo, color}}(\text{Auto})$

Auto	<u>patente</u>	marca	modelo	color
	MBO34L	Ford	Ka	verde
	LDA75K	Toyota	corollaXL	blanco
	ADA89A	Fiat	siena	gris
	LBF78G	Toyota	corollaXL	blanco
	XSA67D	Ford	Ka	rojo

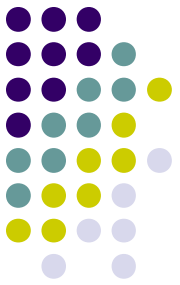
Se eliminan todas salvo una de las tuplas repetidas (Ej, *<Toyota, corollaXL, blanco>*)

R	<u>patente</u>	marca
	MBO34L	Ford
	LDA75K	Toyota
	ADA89A	Fiat
	LBF78G	Toyota
	XSA67D	Ford

Q	marca	modelo	color
	Ford	Ka	verde
	Fiat	siena	gris
	Toyota	corollaXL	blanco
	Ford	Ka	rojo

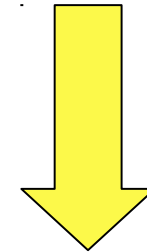
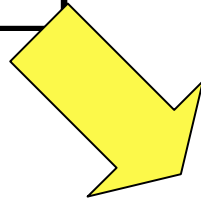
Producto Natural

$$\text{Auto} = R * Q$$



R	<u>patente</u>	marca
	MBO34L	Ford
	LDA75K	Toyota
	ADA89A	Fiat
	LBF78G	Toyota
	XSA67D	Ford

Q	marca	modelo	color
	Ford	Ka	verde
	Toyota	corollaXL	blanco
	Fiat	siena	gris
	Ford	Ka	rojo



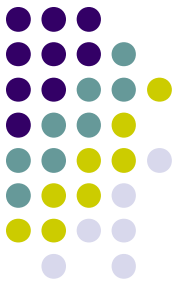
$$\text{Auto} = \sigma_{(R.\text{marca}=Q.\text{marca})}(R \times Q)$$

o bien:

$$\text{Auto} = R * Q$$

Auto	<u>patente</u>	marca	modelo	color
	MBO34L	Ford	Ka	verde
	MBO34L	Ford	Ka	rojo
	LDA75K	Toyota	corollaXL	blanco
	ADA89A	Fiat	siena	gris
	LBF78G	Toyota	corollaXL	blanco
	XSA67D	Ford	Ka	verde
	XSA67D	Ford	Ka	rojo

Descomposición



- Es el reemplazo de una relación $R(A_1, A_2, \dots, A_n)$, por una colección de relaciones R_1, R_2, \dots, R_n obtenidas de las **proyecciones** de R y tal que la relación resultado de los productos naturales de $R_1 * R_2 * \dots * R_n$ tiene el mismo esquema que R .

- Ej: Si tenemos:

$$R_1 = \Pi_{\text{placa, color, modelo}}(\text{Auto})$$

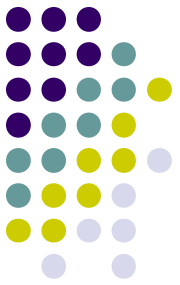
$$R_2 = \Pi_{\text{modelo, marca}}(\text{Auto})$$

...resulta que:

$$R_1 * R_2 = \text{Auto} ?$$

$R * Q \neq \text{Auto}$

¿¿¿ Auto = $R * Q$???



Auto	<u>patente</u>	marca	modelo	color
	MBO34L	Ford	Ka	verde
	MBO34L	Ford	Ka	rojo
	LDA75K	Toyota	corollaXL	blanco
	ADA89A	Fiat	siena	gris
	LBF78G	Toyota	corollaXL	blanco
	XSA67D	Ford	Ka	verde
	XSA67D	Ford	Ka	rojo

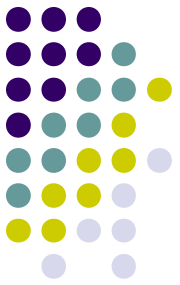
La relación original

¿Qué sucedió aquí?

Auto	<u>patente</u>	marca	modelo	color
	MBO34L	Ford	Ka	verde
	LDA75K	Toyota	corollaXL	blanco
	ADA89A	Fiat	siena	gris
	LBF78G	Toyota	corollaXL	blanco
	XSA67D	Ford	Ka	rojo

$R * Q \neq \text{Auto}$

¿¿¿ Auto = $R * Q$???



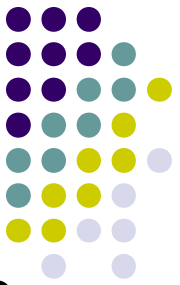
Auto	<u>patente</u>	marca	modelo	color
	MBO34L	Ford	Ka	verde
	MBO34L	Ford	Ka	rojo
	LDA75K	Toyota	corollaXL	blanco
	ADA89A	Fiat	siena	gris
	LBF78G	Toyota	corollaXL	blanco
	XSA67D	Ford	Ka	verde
	XSA67D	Ford	Ka	rojo

La relación original

...En el fondo,
se perdió
información...

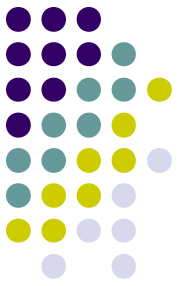
Auto	<u>patente</u>	marca	modelo	color
	MBO34L	Ford	Ka	verde
	LDA75K	Toyota	corollaXL	blanco
	ADA89A	Fiat	siena	gris
	LBF78G	Toyota	corollaXL	blanco
	XSA67D	Ford	Ka	rojo

Descomposición sin Pérdida



- Es la descomposición de una relación R en R_1, R_2, \dots, R_n tal que ***para toda extensión de R*** se tiene que $R = R_1 * R_2 * \dots * R_n$.
- El problema de la concepción de bases de datos relacionales se reduce a la descomposición sin pérdida de las relaciones universales con todos sus atributos, en subrelaciones que no tengan anomalías

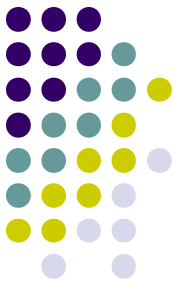
Descomposición sin Pérdida



**¿A dónde nos puede llevar
una mala
descomposición?**

**¿Y una descomposición
insuficiente?**

Anomalías de Actualización

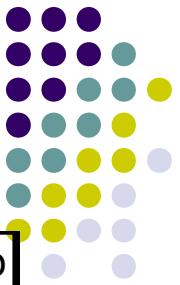


Prof_Depto	Cédula	Nombre_Prof	Fecha_Nac	Código_Depto	Nombre_Depto
	9.980.623	Pedro Pérez	01/06/73	01	Computación
	10.334.890	Luis García	01/06/76	02	Control
	12.334.222	Mario Lobo	01/06/77	01	Computación
	13.434.122	José Rivero	01/06/78	03	Investigación
	13.566.002	Frank Chacón	01/12/78	NULL	NULL
	17.544.672	Miguel Bravo	01/06/84	02	Control
	18.244.670	Andrés León	01/06/85	01	Computación

¿Qué problemas o anomalías se pueden producir en esta relación?

¿Qué “cosas malas” pueden ocurrir?

Anomalías de Actualización



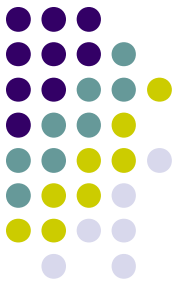
Prof_Depto	Cédula	Nombre_Prof	Fecha_Nac	Código_Depto	Nombre_Depto
	9.980.623	Pedro Pérez	01/06/73	01	Computación
	10.334.890	Luis García	01/06/76	02	Control
	12.334.222	Mario Lobo	01/06/77	01	Computación
	13.434.122	José Rivero	01/06/78	03	Investigación
	13.566.002	Frank Chacón	01/12/78	NULL	NULL
	17.544.672	Miguel Bravo	01/06/84	02	Control
	18.244.670	Andrés León	01/06/85	01	Computación

Anomalías de Inserción: Cada vez que se inserta un profesor es necesario repetir los datos del departamento

Anomalías de Modificación: Cada vez se actualiza un departamento es necesario hacer los cambios en cada una de las tuplas correspondientes a ese departamento

Anomalías de Eliminación: Si se elimina a “José Rivero” asociado a un departamento se pierde la información del departamento

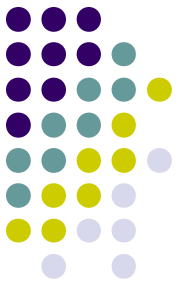
Normalización



¿Qué es normalizar una base de datos?

Es encontrar una descomposición adecuada de la “relación universal” de la base de datos que nos permite cumplir con los criterios de eficacia, ausencia de redundancia, evolución, comprensión, flexibilidad enunciados anteriormente

Normalización



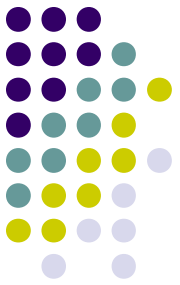
NUEVAMENTE:

¿Qué es normalizar una base de datos?

Es llevar el esquema de la base de datos a alguna de las formas normales . . .

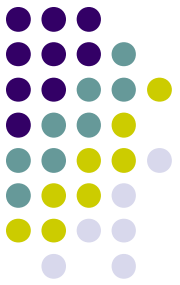
. . . y para eso, necesitamos conocer y comprender el concepto de “dependencia funcional”

Dependencias Funcionales



- Son ***restricciones de integridad*** que permiten conocer que ***relaciones*** existen entre dos o más atributos del mundo real.
- Son propiedades ***inherentes*** al ***contenido semántico*** de los datos, que se han de cumplir para cualquier extensión del esquema de relación.
- Informalmente, Y **depende funcionalmente** de x si ***cada valor*** de x tiene asociado ***siempre el mismo*** valor de Y en una relación R que contiene a x e Y como atributos.

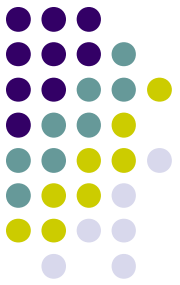
Dependencias Funcionales



El resultado de una consulta cualquiera
(por ejemplo, de un producto entre la tabla profesor y departamento):

Cédula	Fecha_Nac	Sexo	Código_Depto	Nombre_Depto
9.980.623	06/01/73	M	01	Computación
10.334.890	06/01/76	F	01	Computación
17.544.672	06/01/84	M	03	Investigación
12.334.222	06/01/77	M	02	Control
13.566.002	12/01/78	F	02	Control
10.334.890	06/01/76	F	02	Control
12.334.222	06/01/77	M	01	Computación
13.434.122	06/01/78	F	03	Investigación
13.566.002	12/01/78	F	03	Investigación
17.544.672	06/01/84	M	02	Control
18.244.670	06/01/85	M	01	Computación

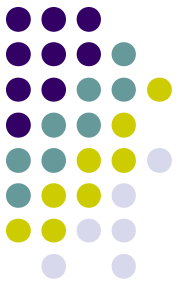
Dependencias Funcionales



Cédula	Fecha_Nac	Sexo	Código_Depto	Nombre_Depto
9.980.623	06/01/73	M	01	Computación
10.334.890	06/01/76	F	01	Computación
17.544.672	06/01/84	M	03	Investigación
12.334.222	06/01/77	M	02	Control
13.566.002	12/01/78	F	02	Control
10.334.890	06/01/76	F	02	Control
12.334.222	06/01/77	M	01	Computación
13.434.122	06/01/78	F	03	Investigación
13.566.002	12/01/78	F	03	Investigación
17.544.672	06/01/84	M	02	Control
18.244.670	06/01/85	M	01	Computación

- ¿Qué características destacan en la información de los profesores?
- ¿Se repite la información de los profesores?
- ¿Que sucede con los atributos Fecha_Nac y Sexo con respecto a la cédula?
- ¿Qué relaciones existen?

Dependencias Funcionales



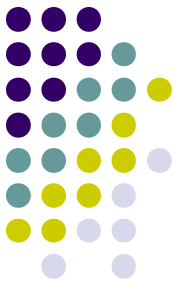
Cédula	Fecha_Nac	Sexo	Código_Depto	Nombre_Depto
9.980.623	06/01/73	M	01	Computación
10.334.890	06/01/76	F	01	Computación
17.544.672	06/01/84	M	03	Investigación
12.334.222	06/01/77	M	02	Control
13.566.002	12/01/78	F	02	Control
10.334.890	06/01/76	F	02	Control
12.334.222	06/01/77	M	01	Computación
13.434.122	06/01/78	F	03	Investigación
13.566.002	12/01/78	F	03	Investigación
17.544.672	06/01/84	M	02	Control
18.244.670	06/01/85	M	01	Computación

¿Y con respecto a la información de los departamentos?

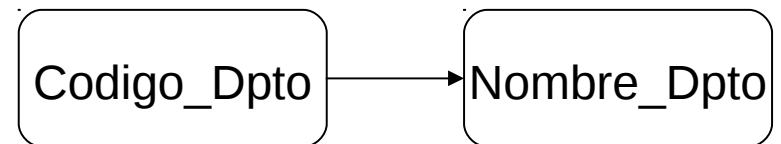
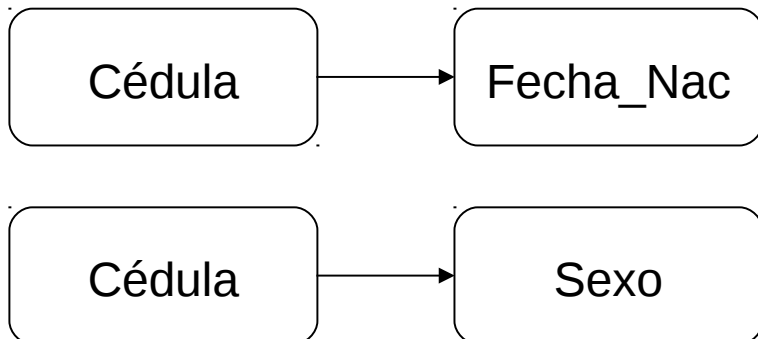
¿Se repite?

¿Qué ocurre con los atributos de las distintas filas?

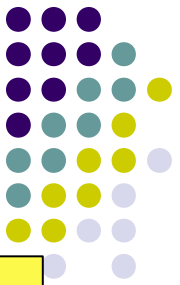
Dependencias Funcionales



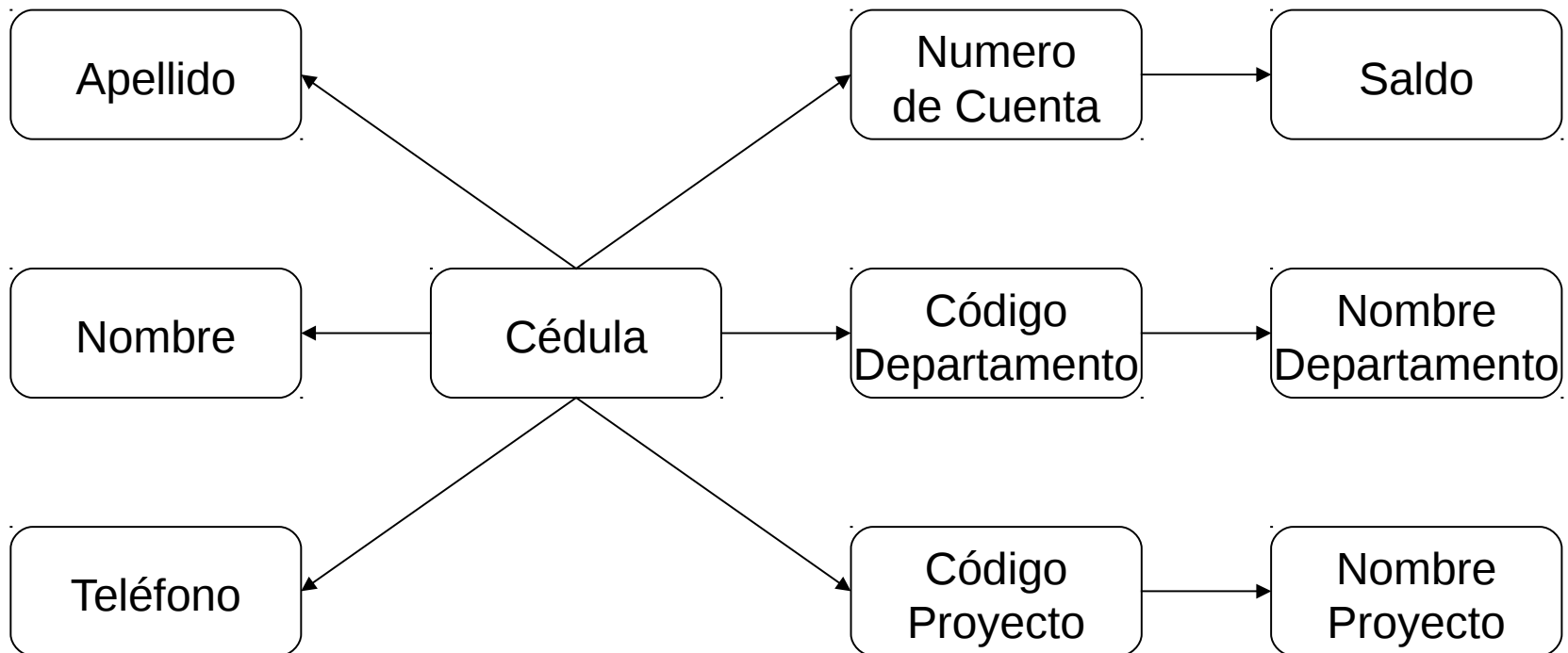
Cédula	Fecha_Nac	Sexo	Código_Depto	Nombre_Depto
9.980.623	06/01/73	M	01	Computación
10.334.890	06/01/76	F	01	Computación
17.544.672	06/01/84	M	03	Investigación
12.334.222	06/01/77	M	02	Control
13.566.002	12/01/78	F	02	Control
10.334.890	06/01/76	F	02	Control
12.334.222	06/01/77	M	01	Computación
13.434.122	06/01/78	F	03	Investigación
13.566.002	12/01/78	F	03	Investigación
17.544.672	06/01/84	M	02	Control
18.244.670	06/01/85	M	01	Computación



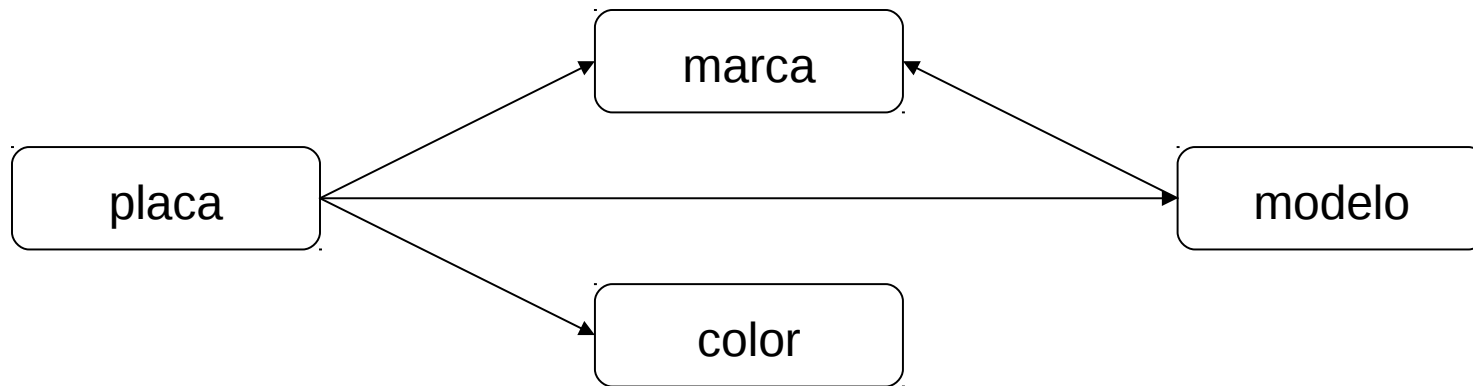
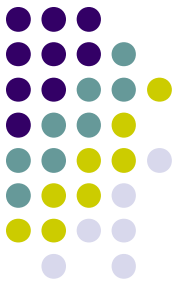
Dependencias Funcionales



Son restricciones de integridad que permiten conocer que interrelaciones existen entre los atributos del mundo real



Caso de Auto



placa → marca
placa → modelo
placa → color
modelo → marca

Dependencias Funcionales



Se viola la dependencia funcional *placa*→*color*

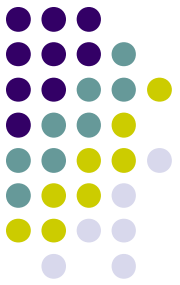
Carro	placa	marca	modelo	color
	MBO34L	Ford	Ka	verde
	MBO34L	Ford	Ka	rojo
	LDA75K	Toyota	corollaXL	blanco
	ADA89A	Fiat	siena	gris
	LBF78G	Toyota	corollaXL	blanco
	XSA67D	Ford	Ka	verde
	XSA67D	Ford	Ka	rojo

Carro	placa	marca	modelo	color
	MBO34L	Ford	Ka	verde
	LDA75K	Toyota	corollaXL	blanco
	ADA89A	Fiat	siena	gris
	LBF78G	Toyota	corollaXL	blanco
	XSA67D	Ford	Ka	rojo

Se cumplen todas las dependencias funcionales

Se viola la dependencia funcional *modelo*→*marca*

Carro	placa	marca	modelo	color
	MBO34L	Ford	Ka	verde
	XXR34L	Chrysler	Ka	rojo
	LDA75K	Toyota	corollaXL	blanco
	ADA89A	Fiat	siena	gris
	LBF78G	Toyota	corollaXL	blanco



Dependencia Funcional

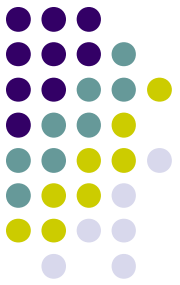
Sea R una relación con atributos (a_1, a_2, \dots, a_n) , y sean X e Y dos subconjuntos de los atributos a_i . Se dice que Y depende funcionalmente de X y se anota

$$X \rightarrow Y$$

Si para todo par de tuplas t_1 y t_2 se cumple

$$t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$$

Dependencia Funcional - Ejemplo



Proveedores (cod-proveedor, nom-proveedor, cod-insumo, precio)

cod-proveedor \rightarrow nom-proveedor
(cod-proveedor, cod-insumo) \rightarrow precio

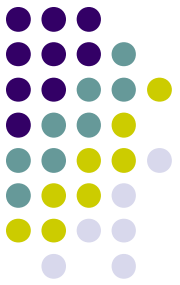
Otra interpretación

cod-proveedor \rightarrow nom-proveedor
cod-insumo \rightarrow precio

Las dependencias funcionales

- sirven para capturar propiedades del mundo real
- dan semántica a las tablas
- definen restricciones

Clave



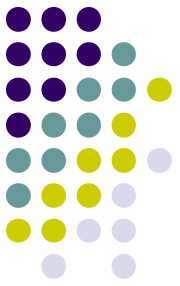
Un subconjunto K de los atributos (a_1, a_2, \dots, a_n) , de una relación R es **clave** si:

- $K \rightarrow (a_1, a_2, \dots, a_n)$, y
- no existe $Y \subset K$ tal que $Y \rightarrow (a_1, a_2, \dots, a_n)$

Nota: si existe $K \subset X$ y $X \rightarrow (a_1, a_2, \dots, a_n)$ y

$$K \rightarrow (a_1, a_2, \dots, a_n),$$

entonces X es superclave o clave candidata



Clausura de un conjunto de DPs

- Deducción de nuevas dfs a partir de un conjunto de dfs

$R(A, B, C, G, H, I)$

dfs:

$A \rightarrow B \quad (1)$

$A \rightarrow C$

$CG \rightarrow H$

$CG \rightarrow I$

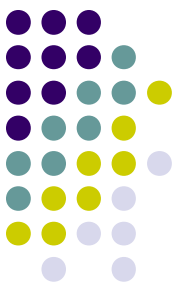
$B \rightarrow H \quad (2)$

si $A \rightarrow B \Rightarrow t_1[A] = t_2[A]$

entonces $t_1[B] = t_2[B] \quad (1)$

y $t_1[H] = t_2[H] \quad (2)$

entonces $A \rightarrow H$



Reglas de Inferencia de DF

⇒ Reflexibilidad:

Si $Y \subseteq X \subseteq R$ entonces $X \longrightarrow Y$ independientemente de F

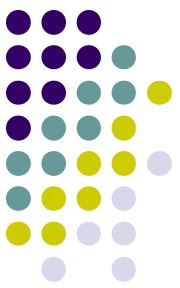
Por ejemplo, $A \longrightarrow A$ y $AB \longrightarrow A$

⇒ Aumentación:

Si $X \longrightarrow Y$ pertenece a F , y $Z \subseteq R$,
entonces $XZ \longrightarrow YZ$.

⇒ Transitividad:

Si $X \longrightarrow Y$ pertenece a F , y también $Y \longrightarrow Z$ pertenece a F ,
entonces $X \longrightarrow Z$.

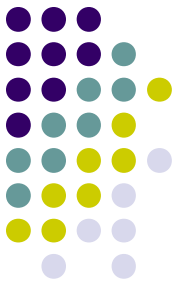


Reglas de Inferencia de DF (2)

A partir de los axiomas básicos se deducen:

- ➡ Unión: Si $X \longrightarrow Y$ y $X \longrightarrow Z$, entonces $X \longrightarrow YZ$.
- ➡ Descomposición: Si $X \longrightarrow YZ$, entonces $X \longrightarrow Y$ y $X \longrightarrow Z$.
- ➡ Seudo-Transitividad: Si $X \longrightarrow Y$ y $WY \longrightarrow Z$,
entonces $XW \longrightarrow Z$.
- ➡ También, todo atributo se determina así mismo $A \longrightarrow A$

Clausura de un Conjunto de Atributos



- Si X es un conjunto de atributos de una relación R , se llama clausura de X , denotado como X^+ , a los atributos determinados funcionalmente por X bajo un conjunto de dfs F .

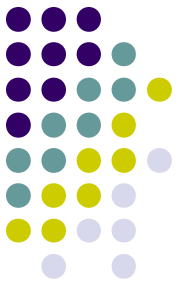
resultado := x

mientras (hay cambios en resultado)

para cada df $y \rightarrow z$ en F

si $y \subseteq \text{resultado} \Rightarrow \text{resultado} := \text{resultado} \cup z$

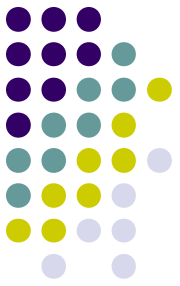
AG⁺



$R(A, B, C, G, H, I)$

$F = \{ A \rightarrow B, \quad A \rightarrow C, \quad CG \rightarrow H, \quad CG \rightarrow I, \quad B \rightarrow H \}$

AG⁺



$R(A, B, C, G, H, I)$

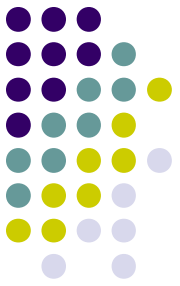
$F = \{ A \rightarrow B, \quad A \rightarrow C, \quad CG \rightarrow H, \quad CG \rightarrow I, \quad B \rightarrow H \}$

1. $resultado = A, G$
2. $resultado = A, G, B$
3. $resultado = A, G, B, C$
4. $resultado = A, G, B, C, H$
5. $resultado = A, G, B, C, H, I$



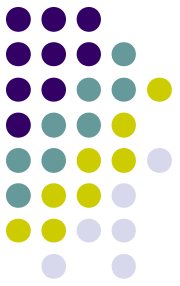
AG es
superclave

Propiedades Deseables de la Descomposición



- No pérdida de información
- Conservar las dependencias funcionales
 - Perder dfs significa perder restricciones.

No pérdida de información



$$R_1 \cap R_2 \rightarrow R_1$$

$$R_1 \cap R_2 \rightarrow R_2$$

Proveedores (cod-proveedor, nom-proveedor, cod-insumo, precio)

cod-proveedor \rightarrow nom-proveedor

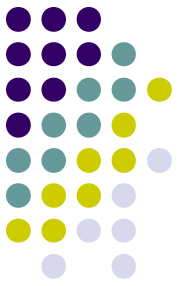
(cod-proveedor, cod-insumo) \rightarrow precio

$R_1 = (\text{cod-proveedor, nom-proveedor})$

$R_2 = (\text{cod-proveedor, cod-insumo, precio})$

$R_1 \cap R_2 = (\text{cod-proveedor}) \rightarrow \text{cod-proveedor, nom-proveedor}$

Conservación de las DFs: ejemplo



- Cuando se dividen las tablas puede ocurrir que se pierdan df.

$(ciudad, calle, codigoPostal)$

$codigoPostal \rightarrow ciudad$

$(ciudad, calle) \rightarrow codigoPostal$

$codigoPostal \rightarrow ciudad$

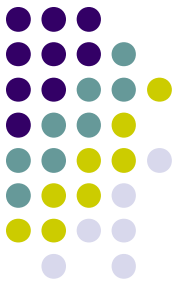


$R_1 = (codigoPostal, ciudad)$

$R_2 = (calle, codigoPostal)$

Si las dependencias se preservan en la división no hay problema.

~~$(ciudad, calle) \rightarrow codigoPostal$~~



Conservación de las DFs

Sea F un conjunto de dfs en el esquema R y R_1, R_2, \dots, R_n una descomposición de R .

La restricción de F a R_i es el conjunto de F_i de todas las dfs en F^+ que incluye solo atributos de R_i .

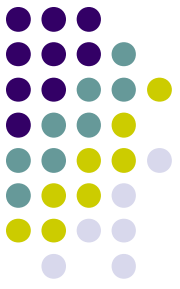
Sea F_1, F_2, \dots, F_n el conjunto de dfs que se prueban sobre R_1, R_2, \dots, R_n respectivamente.

Sea $F' = F_1 \cup F_2 \cup \dots \cup F_n$,

si **$F'^+ = F^+$** entonces **la descomposición conserva las dfs.**

Nota: F' puede ser distinto a F

Conservación de las DFs: Algoritmo



$x \rightarrow y ???$

resultado := x

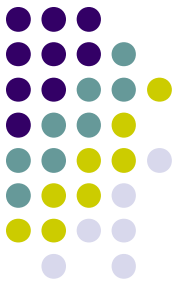
mientras (hay cambios en resultado)

para i=1 hasta n

resultado := resultado $\cup \left((resultado \cap R_i)^+ \cap R_i \right)$

para determinar si F' cubre a F , el algoritmo toma cada $x \rightarrow y$ en F y determina si x^+ calculado sobre F' contiene a y ,

Ejemplo



$$R(A, B, C, D)$$

$$F = \{ A \rightarrow B, B \rightarrow C, C \rightarrow D, D \rightarrow A \}$$

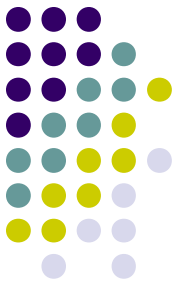
$$R_1 = (A, B) \quad A \rightarrow B, B \rightarrow A,$$

$$R_2 = (B, C) \quad B \rightarrow C, C \rightarrow B$$

$$R_3 = (C, D) \quad C \rightarrow D, D \rightarrow C$$

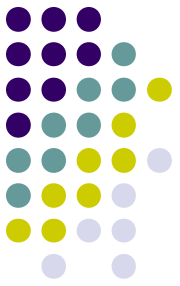
qué pasa con $D \rightarrow A$?

Formas Normales Basadas en Clave Primaria



- Superclave, clave, clave candidata, atributo primo
- Formas normales
 - 1Nf
 - 2NF
 - 3NF
 - BCNF
- Descomposición a 3NF

Superclave, clave, clave candidata, clave primaria

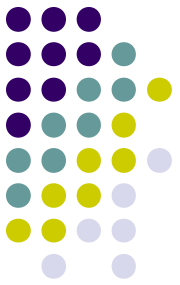


$$R = \{A_1, A_2, \dots, A_n\}$$

$S \subset R$ es superclave si $\forall t_1, t_2$ en ningun posible estado permitido de r de R , $t_1[S] = t_2[S]$

Una clave K es superclave y además si se le quita un atributo cualquiera no deja de ser clave

- *Clave primaria* es un clave elegida arbitrariamente entre las *claves candidatas*
- *Atributo primo*, es un atributo que pertenece a alguna clave candidata



Primera Forma Normal (1NF)

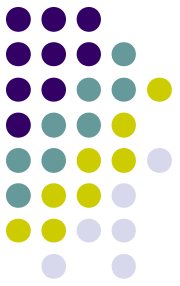
- Una relación **R** está en 1NF si los dominios utilizados sólo contienen valores atómicos o escalares

rut_cliente	deuda
1111-1	(10.000,30/4/93) , (10.000,30/5/93) , (10.000,30/6/93)
9999-9	(20.000,30/4/93) , (10.000,30/5/93)

rut_cliente	monto	fecha_vencimiento
1111-1	10.000	30/4/93
1111-1	10.000	30/5/93
1111-1	10.000	30/6/93
9999-9	20.000	30/4/93
9999-9	20.000	30/5/93

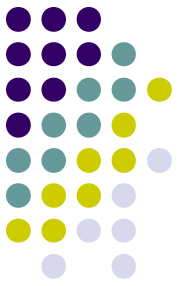
1NF

Segunda Forma Normal (2NF)



- Una relación **R** está en 2FN si está en 1FN si no hay atributos no primos que dependan parcialmente de la clave primaria de **R**.

dependencias total y parcial



$X \rightarrow Y$ es una *dependencia total* si la eliminación de cualquier atributo A de X hace que la dependencia deje de ser válida.

$X \rightarrow Y$ es una *dependencia parcial* si la eliminación de cualquier atributo A de X hace que la dependencia siga siendo válida.

Emp – Proy (dni, # proyecto, horasTrabajadas, nomApellido, nombreProy, lugarProy)

$\{dni, \# proyecto\} \rightarrow horasTrabajadas$

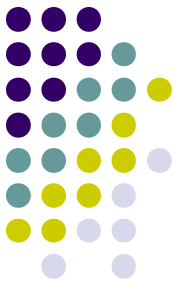
$\{dni, \# proyecto\} \rightarrow nomApellido$

total

parcial

no
esta
en
2NF

Otro ejemplo



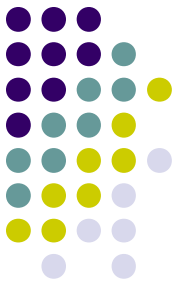
PROVEEDORES(codprov, nomprov, codinsumo, precio)

codprov \rightarrow nomprov \leftarrow parcial

(codprov, codinsumo) \rightarrow precio

no
esta
en
2NF

Llevar a 2NF



Emp – Proy (*dni, # proyecto*, *horasTrabajadas, nomApellido, nombreProy, lugarProy*)

dni, # proyecto → *horasTrabajadas*

dni → *nomApellido*

proyecto → *nombreProy, lugarProy*

Emp – Proy (*dni, # proyecto*, *horasTrabajadas*)

dni, # proyecto → *horasTrabajadas*

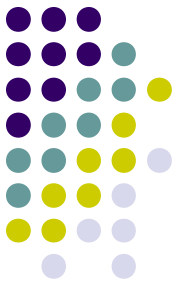
Empleado (*dni, nomApellido*)

dni → *nomApellido*

Proyecto (*proyecto, nombreProy, lugarProy*)

proyecto → *nombreProy, lugarProy*

Hacer que los atributos no primos estén asociados sólo a la parte de la clave primaria de la que dependen totalmente



Dependencia Transitiva

- La 3NF se basa en el concepto de *dependencia transitiva*

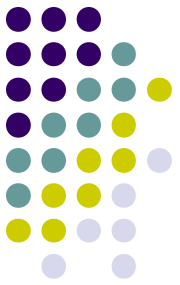
$X \rightarrow Y$ es una *dependencia transitiva* si existe un conjunto de atributos Z que no sea subconjunto de la clave primaria de R y se cumplen que $X \rightarrow Z$ y $Z \rightarrow Y$.

Emp – Depto (*nomApellido*, *dni*, *fechaNac*, *direccion*, *#depto*, *nombreDepto*, *dniGerenteDpto*)

$dni \rightarrow nomApellido, fechaNac, direccion, \#depto$

$depto \rightarrow nombreDepto, dniGerenteDpto$

$dni \rightarrow dniGerenteDpto$ ← **es transitiva**



Tercera Forma Normal (3NF)

- Una relación R está en 3FN si está en 2FN y ningún atributo no primo de R depende transitivamente de la clave primaria.

Una relación o tabla R está en 3FN si está en 2FN, y todas las dependencias funcionales $X \rightarrow A$, con $X \subseteq R$ y $A \in R$ que se cumplen, son de alguno de los siguientes tipos:

- $X \rightarrow A$ es trivial (e.g., la tabla es toda llave).
- X es o contiene una llave (superllave).
- A está contenido en una llave para R .

Ejemplo



Emp – Depto (*nomApellido,dni*, *fechaNac,direccion ,#depto,nombreDepto,dniGerenteDpto*)

dni → *nomApellido,fechaNac,direccion ,#depto*

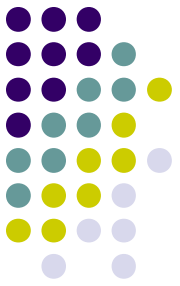
depto → *nombreDepto,dniGerenteDpto*

dni → *dniGerenteDpto* ← **es transitiva**

no
esta
en
3NF

- La diferencia entre 2NF y 3NF es que en 3NF no se admiten dependencias entre dos lados que no estén en la clave
- Todos los atributos no primos dependen de toda la clave (2NF) y sólo de la clave (3NF)

Otro ejemplo

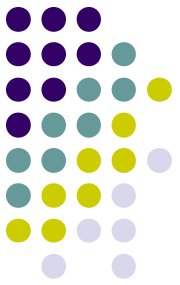


$(\underline{ciudad, calle}, codigoPostal)$

$codigoPostal \rightarrow ciudad$

$(ciudad, calle) \rightarrow codigoPostal$





Ejemplos (1)

1. $R=(dni, \underline{\#cta}, saldo, nyap)$

$dni \rightarrow nyap$

$\#cta \rightarrow saldo, dni$

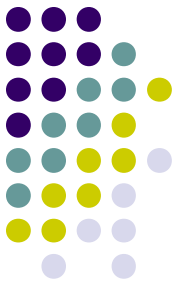
2. $R=(\underline{dni}, \underline{\#cta}, saldo, nyap)$

$dni \rightarrow nyap$

$\#cta \rightarrow saldo$

no
esta
en
3NF

no
esta
en
2NF



Ejemplos (2)

1. $R = (\text{dni}, \underline{\text{\#cta}}, \text{saldo}, \text{nyap})$

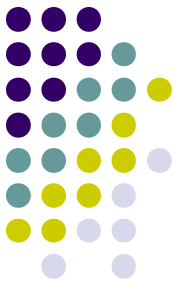
$\text{dni} \rightarrow \text{nyap}$

$\text{\#cta} \rightarrow \text{saldo}, \text{dni}$

2. $R1 = (\text{dni}, \underline{\text{\#cta}}, \text{saldo})$

$R2 = (\underline{\text{dni}}, \text{nyap})$

Ejercicios ...



Lotes (*idPropiedad*, *nMunicipio*, *#lote*, *área*, *precio*, *tasaFiscal*)

idPropiedad → *nMunicipio*, *#lote*, *área*, *precio*, *tasaFiscal*

nombreMunicipio, *#lote* → *idPropiedad*, *área*, *precio*, *tasaFiscal*

nMunicipio → *tasaFiscal*

área → *precio*

idPropiedad → *precio* ????

nMunicipio, *#lote* → *precio* ????

área → *precio*

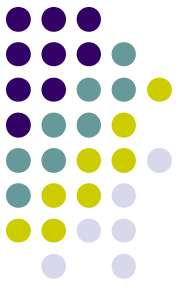
$R_1 = (\underline{\textit{área}}, \textit{precio})$

$R_2 = (\underline{\textit{idPropiedad}}, \textit{nMunicipio}, \textit{\#lote}, \textit{área}, \textit{tasaFiscal})$

idPropiedad → *nMunicipio*, *#lote*, *área*, *tasaFiscal*

nombreMunicipio, *#lote* → *idPropiedad*, *área*, *tasaFiscal*

nMunicipio → *tasaFiscal*



Ejercicios ... cont.

 $R_2 = (\underline{idPropiedad}, nMunicipio, \#lote, \acute{a}rea, tasaFiscal)$

$idPropiedad \rightarrow nMunicipio, \#lote, \acute{a}rea, tasaFiscal$

$nombreMunicipio, \#lote \rightarrow idPropiedad, \acute{a}rea, tasaFiscal$

$nMunicipio \rightarrow tasaFiscal$

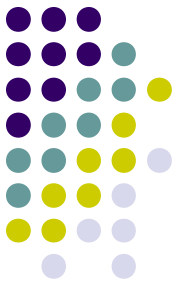
 $nMunicipio \rightarrow tasaFiscal$

$R_3 = (\underline{nMunicipio}, tasaFiscal)$

 $R_4 = (\underline{idPropiedad}, nMunicipio, \#lote, \acute{a}rea)$

$idPropiedad \rightarrow nMunicipio, \#lote, \acute{a}rea$

$nombreMunicipio, \#lote \rightarrow idPropiedad, \acute{a}rea$



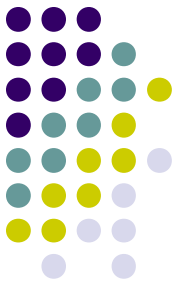
Dependencia Multivaluada

Sea R un esquema de relación formado por los atributos A, B, C . La dependencia multivaluada

$$A \twoheadrightarrow B$$

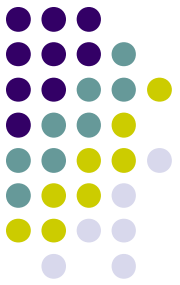
existe si y sólo si el conjunto de valores de B que se obtiene para un par de valores (A, C) depende sólo del valor de A y es independiente de los valores para C .

ejemplo



curso	profesor	texto
C1	P1	T1
C1	P1	T2
C1	P2	T1
C1	P2	T2

curso $\rightarrow\rightarrow$ profesor|texto.



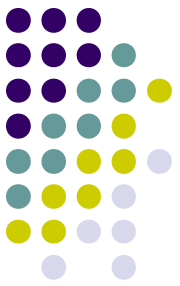
Cuarta Forma Normal (4NF)

Sea R un esquema de relación, R está en 4NF si y sólo si todas las dep. multivaluadas de la forma:

$$X \twoheadrightarrow Y$$

con $X \subseteq R$ e $Y \subseteq R$ se cumple que:

- $X \twoheadrightarrow Y$ es trivial (ej. $R=(X,Y)$)
 - X es superclave de R
-
- Si R esta en 4NF, también esta en BCNF



Llevando a 4NF

(codprod, depto, codinsumo, codprov)

$\text{codprod} \twoheadrightarrow \text{depto}$

$R_1 = (\underline{\text{codprod}}, \text{depto})$

esta en
4NF

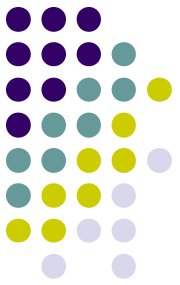
$R_2 = (\underline{\text{codprod}}, \text{codinsumo}, \text{codprov})$

no esta en
4NF

$\text{codinsumo} \twoheadrightarrow \text{codprov}$

$R_3 = (\underline{\text{codprod}}, \text{codinsumo})$

$R_4 = (\text{codinsumo}, \underline{\text{codprov}})$



Propiedades de la descomposición

No pérdida de información

$$(R_1 \cap R_2) \rightarrow\rightarrow (R_1 - R_2)$$

$$(R_1 \cap R_2) \rightarrow\rightarrow (R_2 - R_1)$$

$$R_1 \cap R_2 = \{\text{codprod}\}$$

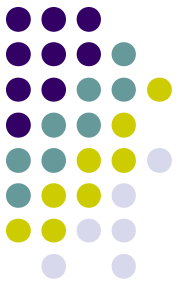
$$R_1 - R_2 = \{\text{depto}\}$$

Sea R_1, R_2, \dots, R_n una descomposición de R , las dep. multivaluadas que están en R que también están en R_i son de la forma:

$$X \rightarrow\rightarrow Y \cap R_i$$

donde $X \subseteq R_i$ y $X \rightarrow\rightarrow Y$ se cumple en R

ejemplo



$$R(A, B, C)$$

$$A \longrightarrow B \text{ y } B \longrightarrow C,$$

$$R_1(A, B)$$

$$R_2(B, C)$$

~~$$R_1(A, B) \text{ y } R_2(A, C)$$~~

$$A \twoheadrightarrow B \text{ y } B \twoheadrightarrow C$$