



**Departamento de
Ciencias de la Computación y Tecnologías de Información
Universidad del Bío-Bío
Sede Chillán**

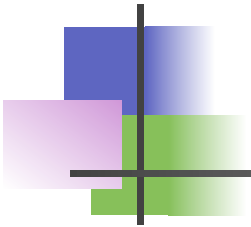
Bases de Datos

Normalización de Esquemas

M^a Angélica Caro Gutiérrez

<http://www.face.ubiobio.cl/~mcaro/>

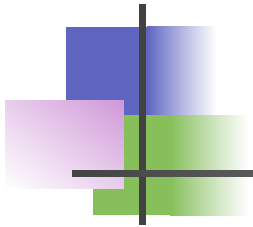
mcaro@ubiobio.cl



Dependencias Funcionales

■ Instancias Legales y Clave Primaria

- Una instancia de una relación R es **legal** si satisface todas las restricciones de integridad sobre R , incluyendo las DFs.
- Examinando las instancias de una relación se puede saber si una DF no se cumple
- Las restricciones de clave primaria son un caso especial de DFs.
 - Los atributos en la clave desempeñan el papel de X y el conjunto de todos los atributos de la relación interpretan el de Y
- La definición de DFs no exige que el conjunto X sea mínimo, sin embargo para que X sea clave este debe ser mínimo.



Dependencias Funcionales

■ Ejemplo Clave Primaria:

- En el esquema Navegantes(**idn**, nombre, categoria, edad) la DF que define la clave primaria es:
 $\text{idn} \rightarrow \text{idn}, \text{nombre}, \text{categoria}, \text{edad}$
- Por cada valor de idn existe un valor único para (idn), nombre, categoria y edad
 - En $\text{idn} \rightarrow \text{idn}, \text{nombre}, \text{categoria}, \text{edad}$, el atributo idn es mínimo
 - En $\text{idn}, \text{nombre} \rightarrow \text{categoria}, \text{edad}$, el conjunto idn, nombre no es minimal, ya que sabemos que $\text{idn} \rightarrow \text{idn}, \text{nombre}, \text{categoria}, \text{edad}$



Dependencias Funcionales

■ Dependencia funcional plena o completa:

- Si el descriptor X es compuesto, por ejemplo, $X(X_1, X_2)$, se dice que Y tiene *dependencia funcional completa o plena* de X , si depende funcionalmente de X , pero no depende de ningún subconjunto del mismo, esto es:
 - $X \rightarrow Y$
 - $X_1 \nrightarrow Y$
 - $X_2 \nrightarrow Y$
- Se representa como $X \Rightarrow Y$, por tanto
$$X \Rightarrow Y \text{ si, y sólo si, } \nexists X' \subset X \mid X' \rightarrow Y$$



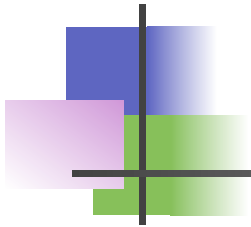
Dependencias Funcionales

■ Dependencia funcional plena o completa:

- Ejemplo 1: PUBLICA(artículo, revista, número, pagin)

Refleja la pág. Inicial en que comienza un artículo en una revista. Un mismo artículo puede aparecer publicado en distintas revistas y en cada una en páginas distintas.

- (articulo, revista, numero) \rightarrow pagin
- articulo \nrightarrow pagin
- revista \nrightarrow pagin
- numero \nrightarrow pagin
- (articulo, revista) \nrightarrow pagin
- (articulo, numero) \nrightarrow pagin
- (revista, numero) \nrightarrow pagin
- por tanto, (articulo, revista, numero) \Rightarrow pagin



Dependencias Funcionales

- **Dependencia funcional plena o completa:**

- Ejemplo 2 :

MATRICULA (Cód_Curso, Cód_Edición, Cód_Estudiante, Nota)

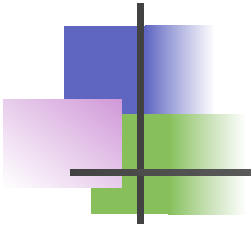
- DF plena:

(Cód_Curso, Cód_Edición, Cód_Estudiante) \Rightarrow Nota

refleja que la nota la obtiene un estudiante en una edición determinada de un curso determinado.

- **Atributo extraño:** son los atributos del determinante de una DF que hacen que ésta no sea plena. También se llaman ajenos. Ejemplo según BD anterior:

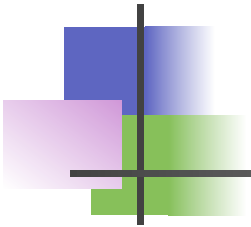
- La DF (Cód_Estudiante, Cód_Curso) \Rightarrow Cód_Programa
- Es no plena y Cód_Estudiante es un atributo extraño.



Dependencias Funcionales

■ Dependencia funcional trivial:

- Una dependencia funcional $X \rightarrow Y$ se dice que es trivial si $Y \subseteq X$.
- Ejemplo:
 - $\text{codLibro} \rightarrow \text{codLibro}$
 - $(\text{articulo}, \text{revista}) \rightarrow \text{revista}$

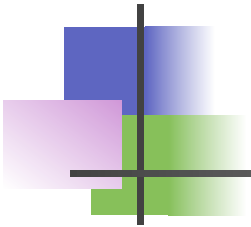


Dependencias Funcionales

■ Dependencia funcional elemental:

- Decimos que una dependencia funcional $X \rightarrow Y$ es **elemental** si *Y es un atributo único* no incluido en X, y no existe X' incluido en X tal que $X' \rightarrow Y$, es decir, la *dependencia funcional elemental* es una *dependencia funcional completa no trivial en que el implicado es un atributo único*.
- Ejemplo:
 - (articulo, revista, numero) \rightarrow pagin

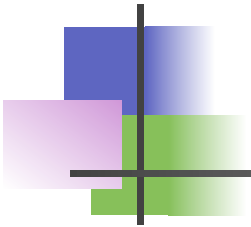
Únicamente las DF elementales **son útiles para la normalización**. El resto de DF no interesan.



Dependencias Funcionales

■ Descriptores equivalentes:

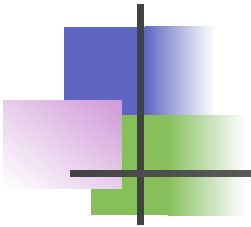
- Se dice que los descriptores X e Y son equivalentes, si se cumple:
 - $X \rightarrow Y$
 - $Y \rightarrow X$
- Lo que se representa por
 - $X \leftrightarrow Y$
- Ejemplos:
 - $\text{codLibro} \leftrightarrow \text{titulo}$
 - $\text{CodSocio} \leftrightarrow \text{Rut}$



Dependencias Funcionales

■ Dependencia funcional transitiva:

- Sea la relación cuyo esquema es $R(X, Y, Z)$ en las que existen las siguientes dfs:
 - $X \rightarrow Y$
 - $Y \rightarrow Z$
 - $Y \nrightarrow X$
- se dice que Z tiene una dependencia funcional transitiva respecto de X a través de Y .
- Si, además $Z \nrightarrow Y$ se dice que la dependencia transitiva es estricta.



Dependencias Funcionales

- **Dependencia funcional transitiva:**

- Ejemplo 1:

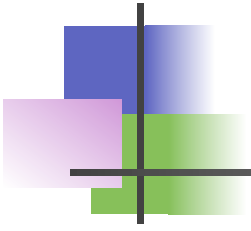
- `LIBRO_EDITORIAL(codLibro, editorial, pais)`

- Donde se tienen las DFs:

- `codLibro → editorial`

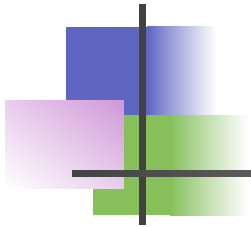
- `editorial → pais`

- entonces también existe la df **`codLibro → pais`**



Unidad 6: Normalización

- Introducción al refinamiento de esquemas
- Dependencias Funcionales (DFs)
- ➡ ■ Razonamiento sobre DFs
- Dependencias Multivaluadas, de Proyección y Unión

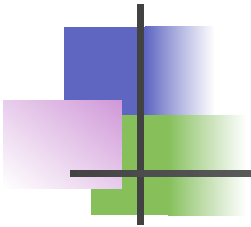


Razonamientos sobre DFs

■ Implicación lógica de dfs y axiomas de Armstrong:

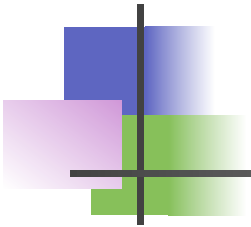
- Sea el esquema de relación $R(A, DF)$, es posible deducir de DF nuevas dfs que sean una consecuencia lógica del conjunto de partida.
- Ejemplo:
 - $PRESTAMOS1(numSocio, nombreSocio, codLibro, fechaPrestamo)$
 - Donde $DF = \{numSocio \rightarrow nombreSocio, nombreSocio \rightarrow numSocio, (numSocio, codlibro) \rightarrow fechaPrestamo\}$
- podemos deducir lógicamente de DF la siguiente df:
 - $(nombreSocio, codlibro) \rightarrow fechaPrestamo$
- La df anterior se cumple para cualquier extensión r de R y decimos que es una consecuencia lógica de DF, lo que expresamos como:

$DF \models (nombreSocio, codlibro) \rightarrow fechaPrestamo$



Razonamientos sobre DFs

- **Implicación lógica de dfs y axiomas de Armstrong:**
 - **Definición:** Dado el esquema de relación $R(A,DF)$, una df d es una consecuencia lógica de DF (expresado $DF \models d$) si d se cumple para cualquier instancia r de R .



Dependencias Funcionales

■ Cierre de un conjunto de dfs:

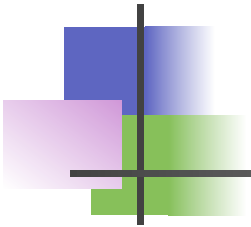
- El cierre de un conjunto de dfs DF (que se denota DF^+) es el conjunto de todas las dfs que son consecuencia lógica de DF:

$$DF^+ = \{X \rightarrow Y \mid DF \models X \rightarrow Y\}$$

- DF será siempre un subconjunto del cierre ($DF \subseteq DF^+$). Por tanto $R(A, DF)$ y $R(A, DF^+)$ definen el mismo esquema de relación.
- El problema es que la definición del cierre de un conjunto de dfs no permite el cálculo del mismo, siendo necesarios unas reglas que faciliten la implicación lógica de dfs (Axiomas de Armstrong).
- Dado un conjunto de dfs DF se dirá que d se deriva de DF, lo que se representa:

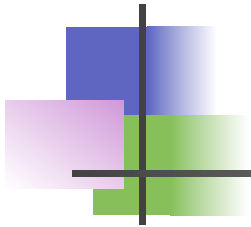
$$DF \vdash d$$

- si d se puede obtener por aplicación sucesiva de dichas reglas a partir de DF (o de un subconjunto de DF)



Dependencias Funcionales

- **Axiomas de Armstrong :**
 - Sean los descriptores X, Y, Z, W , los axiomas de Armstrong establecen:
 - 1. Reflexividad.** Si $Y \subseteq X$, entonces $X \rightarrow Y$ (dependencia trivial)
 - 2. Aumentatividad.** Si $X \rightarrow Y$ y $Z \subseteq W$, entonces $XW \rightarrow YZ$
 - 3. Transitividad.** Si $X \rightarrow Y$ e $Y \rightarrow Z$, entonces $X \rightarrow Z$
 - 4. Proyectividad.** Si $X \rightarrow Y$, entonces $X \rightarrow Y'$ si $Y' \subseteq Y$
 - 5. Unión o aditividad.** Si $X \rightarrow Y$ y $X \rightarrow Z$, entonces $X \rightarrow YZ$
 - 6. Pseudotransitividad.** Si $X \rightarrow Y$ e $YW \rightarrow Z$, entonces $XW \rightarrow Z$



Dependencias Funcionales

■ Ejemplos:

1. Reflexividad.

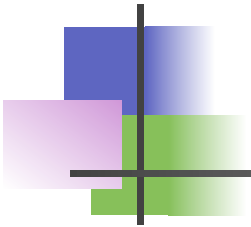
$\{\text{codLibro}, \text{titulo}\} \rightarrow \text{codLibro}$,
ya que $\text{codLibro} \subseteq \{\text{codLibro}, \text{titulo}\}$

2. Aumentatividad.

Si, $\text{codLibro} \rightarrow \text{titulo}$
entonces $\{\text{codLibro}, \text{editorial}, \text{año}\} \rightarrow \{\text{titulo}, \text{editorial}\}$
ya que $\text{editorial} \subseteq \{\text{editorial}, \text{año}\}$

3. Transitividad.

Si $\text{codLibro} \rightarrow \text{editorial}$ y $\text{editorial} \rightarrow \text{año}$
entonces $\text{codLibro} \rightarrow \text{año}$



Dependencias Funcionales

■ Ejemplos:

4. **Proyectividad.**

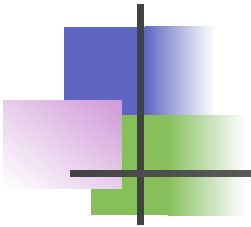
Si $\text{codLibro} \rightarrow \{\text{titulo}, \text{editorial}\}$,
entonces $\text{codLibro} \rightarrow \text{editorial}$
ya que $\text{editorial} \subset \{\text{titulo}, \text{editorial}\}$

5. **Union.**

Si $\text{codLibro} \rightarrow \text{editorial}$ y $\text{codLibro} \rightarrow \text{titulo}$,
entonces $\text{codLibro} \rightarrow \{\text{editorial}, \text{titulo}\}$

6. **Pseudotransitividad.**

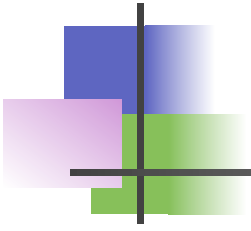
Si $\text{codLibro} \rightarrow \text{titulo}$ y $\{\text{titulo}, \text{numSocio}\} \rightarrow \text{fechaPrestamo}$,
entonces $\{\text{codLibro}, \text{numSocio}\} \rightarrow \text{fechaPrestamo}$



Dependencias Funcionales

■ Ejemplos:

- Dado el esquema de relación $R(A,B,C,D,E; \{A \rightarrow B, C \rightarrow D, D \rightarrow E\})$ demostrar que $AC \rightarrow ABCDE$
 1. $A \rightarrow B$ (dada)
 2. $AC \rightarrow ABC$ (aumentatividad de la anterior por AC)
 3. $C \rightarrow D$ (dada)
 4. $D \rightarrow E$ (dada)
 5. $C \rightarrow E$ (transitividad de 3 y 4)
 6. $C \rightarrow DE$ (unión de 3 y 5)
 7. $ABC \rightarrow ABCDE$ (aumentatividad de 6 por ABC)
 8. $AC \rightarrow ABCDE$ (transitividad de 2 y 7)



Dependencias Funcionales

■ Ejemplos:

- Dado el esquema de relación $R(A,B,C,D; \{A \rightarrow B, B \rightarrow D, A \rightarrow C, A \rightarrow BC\})$ demostrar que $AB \rightarrow ABCD$

1. $A \rightarrow BC$ (dada)
2. $A \rightarrow B$ (dada)
3. $B \rightarrow D$ (dada)
4. $A \rightarrow D$ (transitividad de 2 y 3)
5. $A \rightarrow BCD$ (adición 1 y 4)
6. $AB \rightarrow ABCD$ (aumentatividad por AB)

- ¿AB puede ser la clave principal?

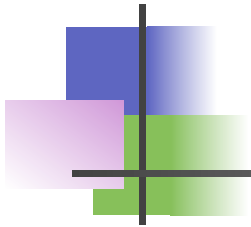


Dependencias Funcionales

■ Ejercicio:

- Dado el esquema de relación $R(A,B,C,D,E)$, demuestre que la dependencia funcional $AB \rightarrow D$ esta en el cierre de $DF = \{AB \rightarrow C, CE \rightarrow D, A \rightarrow E\}$.

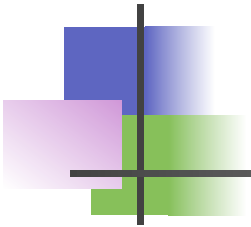
- 1. $A \rightarrow E$ (dada)**
- 2. $AC \rightarrow CE$ (aumentatividad de 1 por C)**
- 3. $CE \rightarrow D$ (dada)**
- 4. $AC \rightarrow D$ (transitividad de 2 y 3)**
- 5. $AB \rightarrow C$ (dada)**
- 6. $AB \rightarrow AC$ (aumentatividad de 5 por A)**
- 7. $AB \rightarrow D$ (transitividad de 4 y 6)**



Dependencias Funcionales

■ Correctitud y completitud de los Axiomas de Armstrong:

- **Correcto.** Si una df $X \rightarrow Y$ se ha derivado por aplicación de los axiomas de Armstrong ($DF \vdash X \rightarrow Y$), dicha df también es consecuencia lógica de DF ($DF \models X \rightarrow Y$) y está contenida en su cierre ($X \rightarrow Y \in DF^+$), es decir, toda df derivada de DF, aplicando los axiomas de Armstrong, se cumple para cualquier extensión r de R .
- **Completo.** Toda df se puede derivar a partir de DF mediante una adecuada aplicación de los axiomas. En otras palabras, aplicando los axiomas de Armstrong a DF podemos encontrar todas las dfs asociadas al esquema de relación $R(A, DF)$



Dependencias Funcionales

■ Nueva definición de DF^+ (cierre de DF)

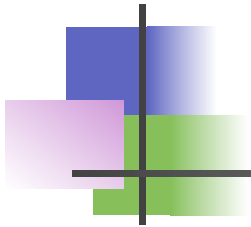
1. DF es un subconjunto de su cierre

$$DF \subseteq DF^+$$

2. Toda $df\ X \rightarrow Y$ derivada de DF mediante la aplicación de los axiomas de Armstrong está en DF^+

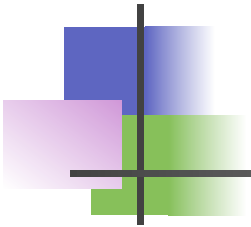
$$DF \vdash X \rightarrow Y \in DF^+$$

3. Ninguna otra df está en DF^+



Dependencias Funcionales

- Trabajar con los axiomas de Armstrong para optimizar (normalizar) esquemas de relación tiene varios **inconvenientes**:
 - Aunque los axiomas de Armstrong facilitan un procedimiento algorítmico para calcular el cierre DF^+ de un conjunto de dependencias, su cálculo consume **mucho tiempo**, ya que, aunque el número inicial de dependencias sea pequeño, el número total de dependencias en el cierre es muy elevado.
 - Para evitar este problema habrá que buscar procedimientos algorítmicos que no estén basados en el cierre de un conjunto de dependencias.
 - Por otro lado, **no todas las dependencias incluidas en el cierre son útiles** en el proceso de diseño de una base de datos, por lo cual se introducirá el concepto de **recubrimiento** o **cobertura irredundante** también llamado **minimal**.



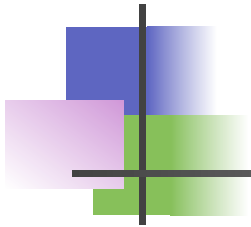
Dependencias Funcionales

■ Definición formal de Superclave

- Dado el esquema de relación $R(A,DF)$, se denomina superclave SK de la relación R a un subconjunto no vacío de A tal que $SK \rightarrow A$ es una consecuencia lógica de DF, siendo por lo tanto un elemento de su cierre, es decir:

$$SK \neq \emptyset \wedge SK \rightarrow A \in DF^+$$

- Esta condición se conoce como propiedad de **unicidad**.
- Significa que una superclave determina a todos los atributos de la relación.
- Ejemplo: Para la relación ya vista $R(A,B,C,D,E; \{A \rightarrow B, C \rightarrow D, D \rightarrow E\})$ el descriptor AC es superclave porque
$$AC \rightarrow ABCDE \quad (\text{todos los atributos})$$



Dependencias Funcionales

■ Definición formal de Clave candidata

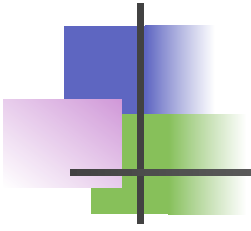
- Para el mismo esquema anterior decimos que K es una clave candidata de R si, además de ser una superclave, no existe ningún subconjunto estricto K' de K tal que K' implique o determine también a A , es decir:

$$K \neq \emptyset \wedge K \rightarrow A \in DF^+ \wedge \nexists K' \subset K \mid K' \rightarrow A$$

- Esta condición se conoce como propiedad de **minimalidad**.
- Significa que una clave candidata tiene como determinante al conjunto mínimo de atributos necesario.
- Ejemplo: Para la misma relación $R(A,B,C,D,E;\{A \rightarrow B, C \rightarrow D, D \rightarrow E\})$
 AC es clave candidata porque es superclave (ya demostrado) y

$$A \twoheadrightarrow ABCDE$$

$$C \twoheadrightarrow ABCDE$$



Dependencias Funcionales

Atributos principales:

- Atributos que forman parte de alguna clave candidata.

Atributos no principales:

- Atributos de una relación que no pertenecen a ninguna de las claves candidatas de la relación.



Dependencias Funcionales

- **Manipulación de DFs**, en base a algoritmos adecuados relacionados con la manipulación de DF. Los principales sirven para:
 1. Determinar si una df $X \rightarrow Y \in DF^+$
 2. Determinar la equivalencia entre dos conjuntos de dfs
 3. Encontrar un recubrimiento minimal de un conjunto de dfs (base para la normalización)
 4. Verificar si un descriptor es clave de un esquema de relación
 5. Obtener todas las claves candidatas de un esquema de relación



Dependencias Funcionales

■ Cierre de un descriptor

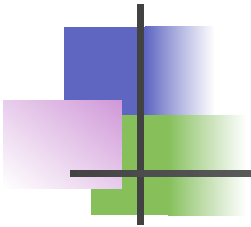
- Dado un esquema de relación $R(A, DF)$, se define el cierre transitivo de un descriptor X respecto al conjunto de DF , que se denota:

$$X^+_{DF}$$

como el conjunto de los atributos de A tales que:

$$X \rightarrow X^+_{DF} \in DF^+$$

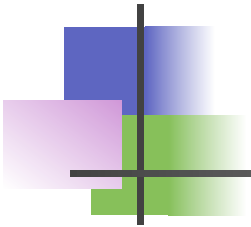
siendo X^+_{DF} máximo en el sentido de que la adición de cualquier atributo vulneraría la condición anterior.



Dependencias Funcionales

■ Algoritmo para el cálculo del cierre de un descriptor

```
1: CierreDescriptor(X, DF)
2:  $X_{DF}^+ = X$ 
3: while sea posible añadir mas atributos a  $X^+$  do
4:   for cada  $df\ Y \rightarrow A \in DF$  do
5:     if  $Y \in X_{DF}^+$  y  $A \notin X_{DF}^+$  then
6:        $X_{DF}^+ = X_{DF}^+ \cup A$ 
7:     end if
8:   end for
9: end while
10: return  $X_{DF}^+$ 
```

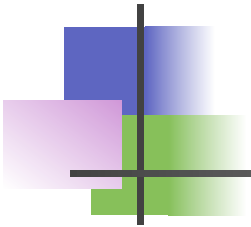


Dependencias Funcionales

■ Ejemplo 1:

- Sea $R = (\{ABCDEFG\}, DF)$
- $DF = \{f1 = AB \rightarrow C, f2 = C \rightarrow A, f3 = BC \rightarrow D, f4 = ACD \rightarrow B, f5 = D \rightarrow EG, f6 = CG \rightarrow BD, f7 = BE \rightarrow C, f8 = CE \rightarrow AG\}$
- $X = BD$
 - $X0 = BD$
 - $X1 = BDEG$ (usando $f5$)
 - $X2 = BDEGC$ (usando $f7$)
 - $X3 = BDEGCA$ (usando $f8$)
 - Como no hay más atributos, éste es el cierre

$$BD^+ = BDEGCA$$



Dependencias Funcionales

■ Ejemplo 2:

Dada la relación $R (\{CE, NE, P, G, CP, C\}, DF)$

con $DF = \{CE \rightarrow NE, NE \rightarrow CE, P \rightarrow CE, G \rightarrow P, (CP, P) \rightarrow G, CE \rightarrow C, P \rightarrow C\}$

hallar el cierre del descriptor (CP, P)

$CP, P \rightarrow CP, P$

iteración 0

$CP, P \rightarrow CP, P, CE, G, C$

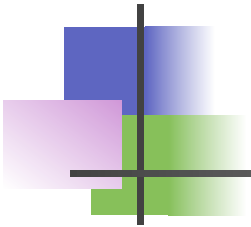
iteración 1

$CP, P \rightarrow CP, P, CE, G, C, NE$

iteración 2

Luego el cierre transitivo del descriptor es:

$(CP, P)^+ = CP, P, CE, G, C, NE$

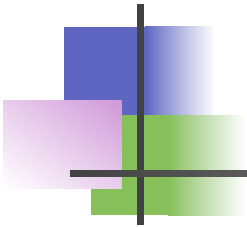


Dependencias Funcionales

■ Cálculo del cierre de un descriptor

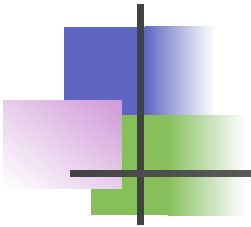
■ Ayuda a:

- Determinar si una dependencia $X \rightarrow Y$ está implicada por un conjunto de dependencias.
- Obtener las claves de un esquema y saber si un descriptor es clave.
- Es la base para calcular la equivalencia de 2 conjuntos de dependencias.
- Calcular la cobertura minimal de un conjunto de dependencias.



Dependencias Funcionales

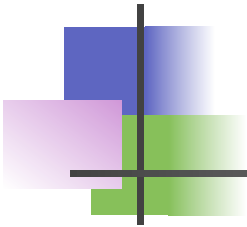
- Comprobar si una dependencia funcional $X \rightarrow Y$ se deriva de un conjunto de dependencias DF equivale a comprobar si $X \rightarrow Y$ pertenece a DF^+
- **Algoritmo para determinar si $DF \vdash X \rightarrow Y$**
 - 1: **dfseDerivadeDF**(df, DF) {df es una dependencia funcional $X \rightarrow Y$ }
 - 2: $X^+_{DF} = \text{CierreDescriptor}(X, DF)$
 - 3: **if** $Y \subset X^+_{DF}$ **then**
 - 4: **return** true {df $\in DF^+$, es decir, $DF \vdash X \rightarrow Y$ }
 - 5: **else**
 - 6: **return** false {df $\notin DF^+$ }
 - 7: **end if**



Dependencias Funcionales

Ejemplo:

- Dada la relación $R (\{CE, NE, P, G, CP, C\}, DF)$
con $DF = \{CE \rightarrow NE, NE \rightarrow CE, P \rightarrow CE, G \rightarrow P,$
 $(CP, P) \rightarrow G, CE \rightarrow C, P \rightarrow C\}$
- Comprobar si la dependencia $NE \rightarrow C$ se deriva de DF .
 - 1: Se calcula el cierre de NE :
 $NE^+ = NE, CE, C$
 - 2: Como C está en el cierre de NE , se cumple que
 $NE \rightarrow C$ pertenece a DF^+ y por tanto, se deriva de DF



Dependencias Funcionales

■ Equivalencia de dos conjuntos de dfs

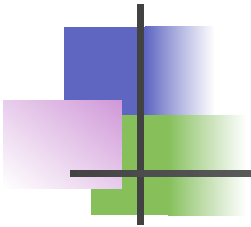
- El problema de la **equivalencia de dos conjuntos de DF** es fundamental en el proceso de normalización, a fin de comprobar si la transformación de un esquema relacional se ha realizado **conservando la semántica**, al menos en lo que a dependencias se refiere.
- Dos conjuntos de dependencias DF_1 y DF_2 son equivalentes si sus cierres son iguales:
$$DF_1^+ = DF_2^+$$
- Para evitar el costo computacional del cálculo de los cierres, se puede comprobar si cada dependencia de DF_1 se encuentra en DF_2 y, viceversa, si cada dependencia de DF_2 se encuentra en DF_1 .



Dependencias Funcionales

■ Equivalencia de dos conjuntos de dfs

- DF_1 y DF_2 son equivalentes, si $DF_1^+ = DF_2^+$
- Podemos verificarlo de la siguiente manera:
 1. Para toda df $X \rightarrow Y \in DF_2$ se cumple que $Y \subset X_{DF_1}^+$.
Significa que toda df $\in DF_2$ está en DF_1 y, por lo tanto, DF_1 es un recubrimiento de DF_2 .
 2. Recíprocamente, para toda df $Z \rightarrow W \in DF_1$, se cumple
 $W \subset Z_{DF_2}^+$
Significa que toda df de DF_1 está en DF_2 y, por lo tanto, DF_1 es un recubrimiento de DF_2 .
- Si se cumple 1) y 2), entonces DF_1 y DF_2 son equivalentes



Dependencias Funcionales

- Ejemplo:

- Dados los siguientes conjuntos de dependencias:

$$DF_1 = \{A \rightarrow B, B \rightarrow A, A \rightarrow C, A \rightarrow D\}$$

$$DF_2 = \{A \rightarrow B, B \rightarrow A, B \rightarrow C, B \rightarrow D\}$$

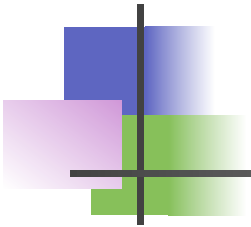
- Las dependencias $A \rightarrow B$ y $B \rightarrow A$ están en ambos conjuntos, por lo que las únicas dependencias de DF_1 que no están en DF_2 son $A \rightarrow C$ y $A \rightarrow D$. Por tanto, debe calcularse el cierre de A con respecto al conjunto DF_2 :

$$A^+_{DF_2} = A, B, C, D$$

- como C y D están contenidos en el cierre, queda demostrado que todas las dependencias de DF_1 están en DF_2^+ , luego DF_2 es un recubrimiento de DF_1 .
- Análogamente, el cierre de B con respecto a DF_1 es:

$$B^+_{DF_1} = B, A, C, D$$

- y por tanto, las dependencias $B \rightarrow C$ y $B \rightarrow D$ de DF_2 están contenidas en DF_1^+ , por lo que DF_1 es un recubrimiento de DF_2 .
- Como conclusión, DF_1 y DF_2 son equivalentes.



Dependencias Funcionales

■ Ejemplo:

- Dados los siguientes conjuntos de dfs:
 - $DF_1 = \{\text{codLibro} \rightarrow \text{titulo}, \text{titulo} \rightarrow \text{codLibro}, \text{codLibro} \rightarrow \text{idioma}, \text{codLibro} \rightarrow \text{editorial}\}$
 - $DF_2 = \{\text{codLibro} \rightarrow \text{titulo}, \text{titulo} \rightarrow \text{codLibro}, \text{titulo} \rightarrow \text{idioma}, \text{titulo} \rightarrow \text{editorial}\}$
- Determinar si DF_1 es equivalente a DF_2



Dependencias Funcionales

■ Solución:

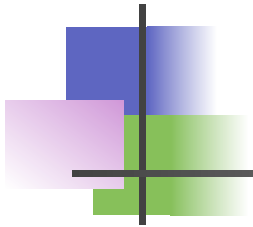
■ DF_1 :

- $codLibro \rightarrow titulo, titulo \rightarrow codLibro$ *están en ambos conjuntos.*
- $codLibro \rightarrow idioma, codLibro \rightarrow editorial$ *no están en DF_2*
- *Entonces se debe calcular el cierre de cod_libro respecto de DF_2 :*
 - *$cod_libro^+ = \{cod_libro, titulo, idioma, editorial\}$*
- *idioma y editorial están contenidos en el cierre, por lo cual se ha demostrado que todas las dependencias de DF_1 están en DF_2^+ , luego DF_2 es recubrimiento DF_1*

■ DF_2 :

- $titulo \rightarrow idioma, titulo \rightarrow editorial$ *no están en DF_1*
- *Entonces se debe calcular el cierre de $titulo$ respecto de DF_1 :*
 - *$titulo^+ = \{cod_libro, titulo, idioma, editorial\}$*
- *idioma y editorial están contenidos en el cierre, por lo cual se ha demostrado que todas las dependencias de DF_2 están en DF_1^+ , luego DF_1 es recubrimiento DF_2*

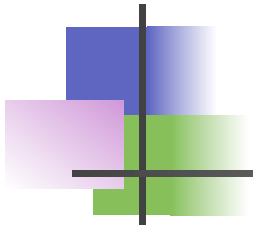
■ Es decir DF_1 es equivalente a DF_2



Dependencias Funcionales

■ Recubrimiento minimal:

- De todos los posibles conjuntos equivalentes a un conjunto dado de dependencias, hay un subconjunto de ellos que cumplen la propiedad que no contienen ninguna **dependencia redundante**, por lo que se llaman conjuntos irredundantes o minimales.
- Los algoritmos de normalización y los de cálculo de claves candidatas parten siempre de recubrimientos minimales.

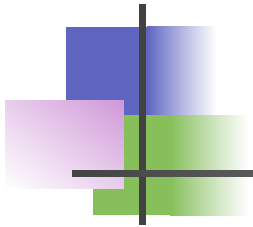


Dependencias Funcionales

■ Recubrimiento minimal:

- El recubrimiento minimal de un conjunto de dfs DF asociadas a un conjunto de atributos A , es un subconjunto de dfs elementales^a de DF tal que cumpla:
 1. Ninguna de las dfs elementales de DF es redundante, es decir, si se elimina cualquiera de las dfs de DF , el nuevo conjunto de dfs DF' no es equivalente a DF .
 2. Todas las dfs entre los atributos de A están en DF^+ .
- **df redundante**: Una df $d \in DF$ se dice que es redundante si puede derivarse de $\{DF - d\}$ mediante los axiomas de Armstrong.
- **atributo extraño**: Dada la df $X \rightarrow Y \in DF$, un atributo $A \in X$ se dice que es extraño en la dependencia si la df $(X - A) \rightarrow Y \in DF^+$.

^a dependencia elemental es aquella, que tiene un único implicado, siendo además plena y no trivial



Dependencias Funcionales

■ Ejemplo de recubrimiento minimal:

- Sea el esquema de relación

LIBRO(A = {codLibro, isbn, editorial, pais},

DF = {codLibro \rightarrow isbn, isbn \rightarrow codLibro, isbn \rightarrow (editorial, pais),
codLibro \rightarrow editorial, editorial \rightarrow pais})

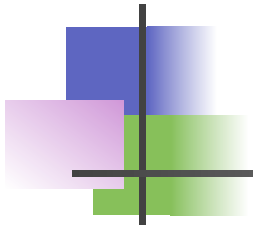
- Los conjuntos

- DF1 = {codLibro \rightarrow isbn, isbn \rightarrow codLibro, codLibro \rightarrow editorial,
editorial \rightarrow pais}

- DF2 = {codLibro \rightarrow isbn, isbn \rightarrow codLibro, isbn \rightarrow editorial,
editorial \rightarrow pais}

son recubrimientos minimales de DF.

- Esto implica que puede haber más de un recubrimiento minimal de un mismo conjunto de dfs, es decir, el recubrimiento de un conjunto de dfs no es único. Además los conjuntos pueden no tener el mismo número de dfs y por lo tanto alguno de ellos puede ser mínimo



Dependencias Funcionales

■ Ejemplo de recubrimiento minimal:

- Sea el esquema de relación

LIBRO($A = \{\text{codLibro}, \text{isbn}, \text{editorial}, \text{pais}\}$, $DF = \{\text{codLibro} \rightarrow \text{isbn}, \text{isbn} \rightarrow \text{codLibro}, \text{isbn} \rightarrow (\text{editorial}, \text{pais}), \text{codLibro} \rightarrow \text{editorial}, \text{editorial} \rightarrow \text{pais}\}$)

- Se tiene que:

- $DF3 = \{\text{codLibro} \rightarrow \text{isbn}, \text{isbn} \rightarrow \text{codLibro}, \text{isbn} \rightarrow \text{pais}, \text{isbn} \rightarrow \text{editorial}, \text{editorial} \rightarrow \text{pais}\}$

No es recubrimiento minimal ya que $\text{isbn} \rightarrow \text{país}$ es redundante

- y

- $DF4 = \{\text{codLibro} \rightarrow \text{isbn}, \text{isbn} \rightarrow \text{editorial}, \text{editorial} \rightarrow \text{pais}\}$

No es recubrimiento minimal ya que $\text{isbn} \rightarrow \text{codLibro}$ falta y no se puede deducir de las demás.



Dependencias Funcionales

■ Algoritmo de Ullman y Atkins:

Entrada: DF (conjunto de dependencias elementales)

Salida: H (recubrimiento minimal de DF)

Proceso :

1) Eliminación de atributos extraños.

1.1) Repetir para cada dependencia $X \rightarrow B$ de DF:

1.1.1) $L = X$

1.1.2) Repetir para cada atributo A de X:

Si $B \in (L - A)^+$ entonces $L = L - A$

1.1.3) Reemplazar $X \rightarrow B$ por $L \rightarrow B$

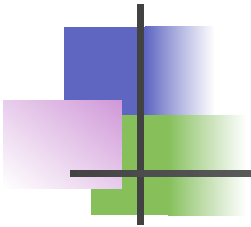
2) Eliminación de dependencias redundantes.

2.1) $H = DF$

2.2) Repetir para cada dependencia $X \rightarrow A$ de DF:

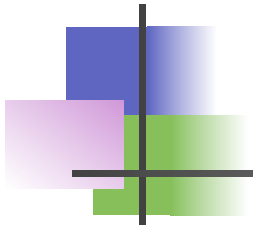
$G = H - \{ X \rightarrow A \}$

Si A pertenece a X_G^+ entonces $H = G$



Dependencias Funcionales

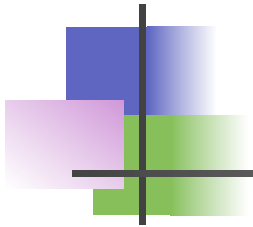
- La utilización de recubrimientos irredundantes o minimales tiene **dos objetivos**:
 1. **Reducir la complejidad algorítmica** al disminuir el número de dependencias de partida), y
 2. **Minimizar el número de restricciones de integridad** que han de ser mantenidas en la base de datos.
- Por ambas razones, debe ser un *objetivo de diseño conseguir que el número de dependencias y el número atributos involucrados sean mínimos*.
- Además, existe **otro objetivo de diseño**, que es aún más importante:
 - *que las dependencias resultantes tengan un significado claro para los usuarios.*
 - Este problema no puede ser resuelto con la teoría de la normalización, ya que realiza transformaciones algorítmicas de tipo sintáctico que pueden conducir a dependencias y a esquemas de relación **absurdos** desde el punto de vista del usuario.



Dependencias Funcionales

■ **Determinación si un descriptor es clave de una relación:**

- Dada una relación $R(A, DF)$, se trata de comprobar si un descriptor X es una clave
 1. Se calcula X^+_{DF} de X
 2. Si $X^+_{DF} \rightarrow A$, entonces X es una superclave
 3. Si $X^+_{DF} \nrightarrow A$, entonces X no es una superclave
- Si X es una superclave, entonces se buscan todos los subconjuntos X'_i de X .
- Si algún $X'^+_{DF} \rightarrow A$, entonces X NO es una clave.
- En caso contrario X es una clave candidata.



Dependencias Funcionales

■ Ejemplo de una clave candidata:

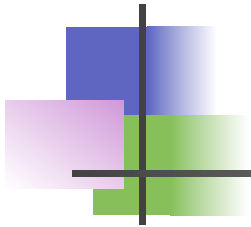
- Sea la relación

PRESTAMOS(numSocio, nombreSocio, codLibro,
fechaPrestamo, editorial, pais)

- con las siguientes dfs

DF = {numSocio → nombreSocio, nombreSocio →
numSocio, codLibro → editorial, editorial →
pais, (numSocio, codLibro) → fechaPrestamo}

- Verificar si el descriptor (numSocio, codLibro) es una clave candidata.



Dependencias Funcionales

■ Solución:

- Calculamos el cierre de (numSocio, codLibro) respecto de DF:
 - {numSocio, codLibro, nombreSocio, editorial, pais, fechaPrestamo}
- Como el cierre coincide con el conjunto de atributos de la relación, entonces (numSocio, codLibro) es **super clave**.
- Cierre de numSocio es {numSocio, nombreSocio}
- Cierre de codLibro es {codLibro, editorial, pais}
- Ninguno coincide con el conjunto de atributos de la relación por lo tanto (numSocio, codLibro) es **clave candidata**.



Dependencias Funcionales

■ Ejemplo 2:

- Dado el esquema de relación $R(AT, DF)$

Con $AT = \{A, B, C, D, E, F\}$ y $DF = \{A \rightarrow B; B \rightarrow A; C \rightarrow E; E \rightarrow F; (A, C) \rightarrow D\}$

- ¿Es (A, C) una clave candidata?
- Como $(A, C)^+_{DF} = (A, C, B, E, D, F) = AT \Rightarrow (A, C)$ es una superclave.
- Además, como $A^+_{DF} = (A, B) \neq AT$ y $C^+_{DF} = (C, E, F) \neq AT \Rightarrow (A, C)$ es una clave candidata.



Dependencias Funcionales

- Dado un esquema de relación $R(A, DF)$, siendo DF un **recubrimiento irredundante**, los pasos para **calcular sus claves candidatas** son:
 - **Paso 1: Eliminación de atributos independientes.**
 - Se eliminan de R todos los atributos independientes (que no forman parte de ninguna dependencia) obteniendo una relación R_{si} .
 - **Paso 2: Eliminación de descriptores equivalentes.**
 - Por cada grupo de descriptores equivalentes ($X \leftrightarrow Y \dots$), se elige uno (por ejemplo X), eliminando las dependencias de equivalencia anteriores de DF y sustituyendo en las dependencias restantes los descriptores eliminados (por ejemplo Y) por el atributo que se ha elegido del grupo (X en este caso).
 - Se obtiene así una relación R_{sie} .
 - Cuando, como resultado de este paso, las relaciones no tienen dependencias, los atributos de las mismas son independientes:
 - Ejemplo: $R(A, B; \emptyset)$ implica que los atributos A y B son independientes.



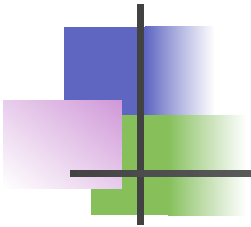
Dependencias Funcionales

- **Paso 3:** *Determinación de un descriptor (en el que no haya implicados) que sea clave de R_{sie} .*
 - Los atributos de una relación R_{sie} que son implicantes pero no implicados son parte de la clave, tomamos estos atributos y con ellos formamos una clave posible (K_p).
 - Si no hay ningún otro implicante que, a la vez, sea implicado, K_p es una clave y se va al paso 5.
 - En caso contrario, se realiza el paso 4.



Dependencias Funcionales

- **Paso 4:** *Determinación de un descriptor clave de R_{sie} (en el que puede haber implicados siempre que sean también implicantes).*
 - Si es posible, se obtiene una partición R'_{sie} eliminando de R_{sie} todos aquellos atributos que entran en K_p^+ y que no forman parte de otras dependencias funcionales, distintas a las que han servido para calcular K_p^+ .
 - En R'_{sie} se obtiene una clave provisional K'_p con los implicantes que estaban también en K_p añadiendo un nuevo implicante que, a su vez, sea también implicado. Si K'^+_p contiene todos los atributos de R'_{sie} es una clave, en caso contrario se añade un nuevo descriptor hasta obtener una clave.
 - Se repite esta operación porque puede haber más claves.
 - Una vez obtenidas las claves de R'_{sie} se hace la unión de cada una de ellas con la clave obtenida en el paso 3 para obtener así las claves de R_{sie} .
 - Si no fuese posible obtener la partición R'_{sie} se actuaría de la misma manera que se acaba de explicar, pero con R_{sie} .



Dependencias Funcionales

- **Paso 5:** *Tratamiento de atributos independientes para obtener una clave de la relación original.*
 - A las claves de R_{sie} obtenidas en los pasos 3 o 4 se añaden los atributos independientes obtenidos en el paso 1 (o en el 2).
- **Paso 6:** *Tratamiento de descriptores equivalentes.*
 - Cuando en el paso 2 se han obtenido descriptores equivalentes habrá que obtener todas las claves, sustituyendo en las claves obtenidas en el paso 5 (si hubiese atributos independientes) o en los pasos 3 o 4 (si no los hubiera), los descriptores por sus equivalentes.
 - De esta forma se obtienen todas las claves candidatas.



Dependencias Funcionales

- ***Cálculo Claves candidatas, Ejemplo 1:***

Sea el esquema de relación:

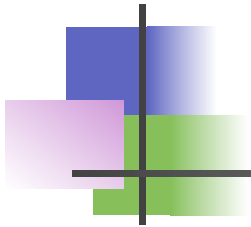
$R(\{A,B,C,D,E,F,G,H,I,J\}; \{AB \rightarrow C, C \rightarrow AB, E \rightarrow D, D \rightarrow E, E \rightarrow F, F \rightarrow E, ABD \rightarrow G, CF \rightarrow H\})$

- **Paso 1**

- Los atributos **I y J son independientes** porque no forman parte de ninguna DF, luego, en este primer paso se eliminan de la relación:
- $R_{si}(\{A,B,C,D,E,F,G,H\}, \{AB \leftrightarrow C, D \leftrightarrow E \leftrightarrow F, ABD \rightarrow G, CF \rightarrow H\})$

- **Paso 2**

- Existen **dos grupos de descriptores equivalentes**:
 - a) **AB y C**
 - b) **D, E y F**
- Del grupo a) nos quedaríamos, por ejemplo, con C y del grupo b) con D (eliminaríamos, por tanto, AB, E y F); la relación resultante sin equivalencias sería:
- $R_{sie}(\{C, D, G, H\}; \{CD \rightarrow G, CD \rightarrow H\})$



Dependencias Funcionales

- ***Cálculo Claves candidatas, Ejemplo 1:***

- **Paso 3**

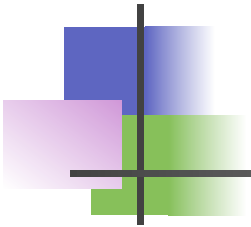
- En la R_{sie} anterior, CD es el único implicante, pero no implicado, luego una K_p sería CD , como el resto son sólo implicados, CD es clave de R_{sie} (no haría falta hallar el cierre de CD). Pasaríamos al paso 5.

- **Paso 5**

- Si a CD le añadimos los atributos independientes I y J tenemos $CDIJ$ que es la clave de R .

- **Paso 6**

- Los descriptores equivalentes eran: $AB \leftrightarrow C$ y $D \leftrightarrow E \leftrightarrow F$
- La clave $CDIJ$ genera las siguientes claves candidatas de R :
 $\{C|AB\}\{D|E|F\}IJ$
- En total, son 6 claves: $CDIJ, CEIJ, CFIJ, ABDIJ, ABEIJ$ y $ABFIJ$



Dependencias Funcionales

- ***Cálculo Claves candidatas, Ejemplo 2:***

Sea el esquema de relación:

$R(\{A,B,C,D,E,F\}; \{AB \rightarrow C, DE \rightarrow F, F \rightarrow D\})$

- **Paso 1**

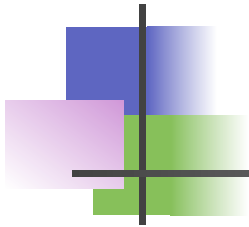
- No hay atributos independientes $\Rightarrow R_{si} = R$

- **Paso 2**

- No hay descriptores equivalentes $\Rightarrow R_{sie}(\{A,B,C,D,E,F\}; \{AB \rightarrow C, DE \rightarrow F, F \rightarrow D\})$

- **Paso 3**

- $K_p = ABE$ y $K_p^+ = ABCE$; luego K_p no es clave, por lo que iríamos al paso 4.



Dependencias Funcionales

■ ***Cálculo Claves candidatas, Ejemplo 2:***

■ Paso 4

- Obtenemos una nueva relación R'_{sie} eliminando de R_{sie} los atributos A B C que forman la primera DF (no eliminamos E porque en la dependencia de la que forma parte aparecen D y F que no están en R_{sie}) y nos queda:

$R'_{sie} (\{DEF\}; \{DE \rightarrow F, F \rightarrow D\})$

- Formaríamos una clave provisional K'_p con E que es sólo implicante (y, por tanto, está en R_{sie}), añadiendo un descriptor implicante e implicado, por ejemplo F:
- $K'_p = EF$ y $K'^+_p = EFD$; luego EF es una clave de la partición R'_{sie}
- Otra clave sería ED. Por tanto, las claves de R_{sie} serían:

ABEF

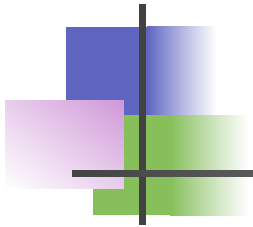
ABED

■ Paso 5

- Como en el paso 1 no hubo atributos independientes, las claves son ABEF y ABED.

■ Paso 6

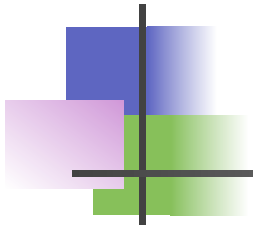
- Como tampoco hubo descriptores equivalentes en el paso 2, las claves son **ABEF y ABED**.



Dependencias Funcionales

■ Teoría de la Normalización de Esquemas Relacionales:

- Dado un conjunto A de atributos y un conjunto D de dependencias entre los atributos, que puede considerarse que constituyen un esquema de relación $R(A,D)$ (esquema origen), se trata de transformar estos conjuntos de partida, en un conjunto de n esquemas de relación $\{R_i(A_i, D_i)\}_{i=1}^n$ (esquemas resultantes) tales que cumplan unas determinadas condiciones.
- Se trata de buscar un conjunto de esquemas R_i que sean **equivalentes** a R y **mejores** que R. Son tres las propiedades que han de cumplir los esquemas de relación R_i para ser **equivalentes** y **mejores** que R; las cuales son:
 1. Conservación de la información
 2. Conservación de las dependencias (*dfs*)
 3. Mínima redundancia de los datos (normalización de las relaciones) (*sólo las basadas en dfs*)



Dependencias Funcionales

■ Continuación:

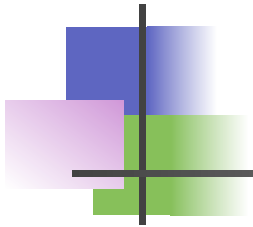
- Si se cumplen las dos primeras propiedades, es decir, la transformación de R en R_i se hace sin pérdida de información y sin pérdida de dfs se dice que:

$\{R_i\}$ es equivalente a R

- Y si todas las relaciones del esquema relacional resultante $\{R_i\}$ están en formas normales más avanzadas que el esquema origen R (hay menos redundancia), se dice que:

$\{R_i\}$ es mejor que R

- Otros objetivos que debe cumplir el conjunto de relaciones resultantes:
 1. Minimización de dfs
 2. Minimización de esquemas resultantes



Dependencias Funcionales

■ Conservación de la Información:

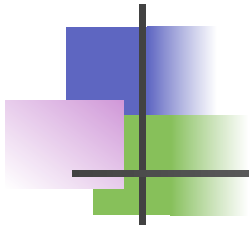
- Para que se cumpla esta propiedad son precisas dos condiciones:

1. **Conservación de los atributos**, es decir,

$$\bigcup_{i=1}^n A_i = A$$

2. **Conservación del contenido de la tuplas.** Para toda relación r de R la reunión natural (join natural) de las relaciones resultantes ha de producir la relación origen r , es decir,

$$*_{i=1}^n r_i = r$$



Dependencias Funcionales

■ Pérdida de Información:

LIBROS

CodLibro	Editorial	Pais
9030	Rama	España
9040	Rama	España
9110	Paraninfo	España
9234	Anaya	España
9567	Addison Wes	EE.UU

LIBROS_NUEVA

CodLibro	Pais
9030	España
9040	España
9110	España
9234	España
9567	EE.UU

EDITORIAL

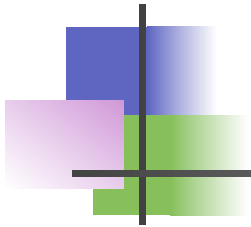
Editorial	Pais
Rama	España
Paraninfo	España
Anaya	España
Adisson Wes	EE.UU

LIBROS_NUEVA * EDITORIAL

CodLibro	Editorial	Pais
9030	Rama	España
(*)9030	Paraninfo	España
(*)9030	Anaya	España
9040	Rama	España
(*)9040	Paraninfo	España
⋮	⋮	⋮

(*) tuplas espúreas

o tuplas erróneas, son las tuplas adicionales que no estaban en la relación original

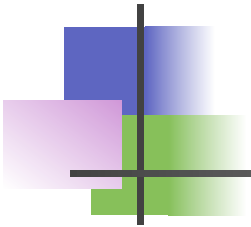


Dependencias Funcionales

■ Conservación de las Dependencias:

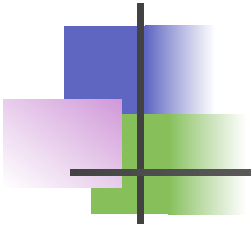
Podemos decir que un esquema de relación $R(A, DF)$ se ha transformado en un conjunto de esquemas $R_i(A_i, DF_i)$ sin pérdida de dependencias si:

$$\left(\bigcup_{i=1}^n DF_i \right)^+ = DF^+$$



Dependencias Funcionales

- La **tercera propiedad** que debe cumplir el conjunto $\{R_i\}$ de esquemas resultantes en un proceso de descomposición, es que estas relaciones alcancen un **nivel de normalización superior al del esquema origen R** , a fin de eliminar en lo posible las redundancias y, por tanto, las anomalías de actualización.
- Se dice que un esquema de relación está en una determinada forma normal, si satisface un cierto conjunto específico de restricciones.
- Cuanto más alta sea la forma normal en la que se encuentran los esquemas de relación, menores serán los problemas que aparecen en el mantenimiento de la BD.

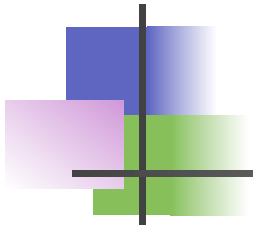


Dependencias Funcionales

Definición Formal de las tres primeras Formas Normales

■ Primera Forma Normal:

- Es una restricción inherente al modelo relacional
- Prohíbe los atributos multivaluados, compuestos o combinaciones
- El Dominio de un atributo debe incluir sólo valores atómicos



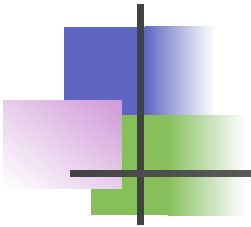
Dependencias Funcionales

■ Segunda Forma Normal:

- Está en 1FN, y
- Cada atributo no principal tiene dependencia funcional completa respecto de cada una de las claves.

■ Ejemplo

- PRESTAMOS(numSocio, nombreSocio, codLibro, fechaPrestamo, editorial, país) con las siguientes dfs:
DF = {NumSocio \leftrightarrow NombreSocio,
(numSocio, codLibro) \rightarrow fechaPrestamo, editorial \rightarrow pais}
claves candidatas: (numSocio, codLibro, editorial) y
(nombreSocio, codLibro, editorial)
- \Rightarrow la relación PRESTAMOS no está en 2FN, pues existe la df editorial \rightarrow pais y el atributo principal editorial es parte de las claves candidatas lo que viola la restricción 2.



Dependencias Funcionales

- La 2FN **no se cumple** cuando:
 - Algún atributo no principal, depende funcionalmente de algún subconjunto de una clave.
- **Siempre están** en 2FN las relaciones:
 - Binarias.
 - Con todas las claves simples, es decir, que contienen un sólo atributo.
 - Que todos los atributos son **principales**, es decir, forman parte de alguna clave.



Dependencias Funcionales

■ Tercera Forma Normal:

- Una relación está en 3FN si:
 1. está en 2FN
 2. Todo atributo que no pertenece a alguna clave candidata depende de la clave y sólo de la clave candidata. Esta restricción también se puede especificar indicando que ningún atributo no principal depende transitivamente de ninguna clave de la relación.

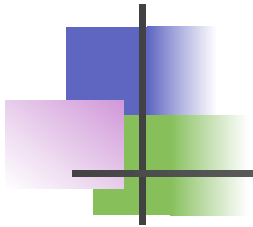
■ Ejemplo:

- LIBROS(codLibro, editorial, país)
 $DF = \{codLibro \rightarrow editorial, editorial \rightarrow pais\}$ La clave de la relación LIBROS es codLibro.
- \Rightarrow LIBROS no está en 3FN, ya que país depende transitivamente de codLibro a través de editorial



Dependencias Funcionales

- La 3FN **no se cumple** cuando:
 - Existen atributos no principales que dependen funcionalmente de otros atributos no principales.
- **Siempre están** en 3FN las relaciones:
 - Binarias.
 - En las que todos los atributos son principales.
 - Que tienen un único atributo no principal.



Dependencias Funcionales

■ Forma Normal de Boyce y Codd (FNBC):

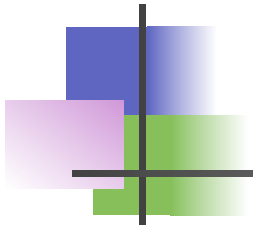
- Se dice que una relación está en FNBC si, y sólo si, todo determinante es una clave.

PRESTAMO1(numSocio, nombreSocio, codLibro, fechaPrestamo)

DF = {numSocio \rightarrow nombreSocio, nombreSocio \rightarrow numSocio,
(numSocio, codLibro) \rightarrow fechaPrestamo}

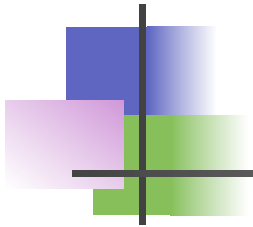
claves candidatas (numSocio, codLibro) y
(nombreSocio, codLibro)

- PRESTAMO1 está en 1FN, 2FN y 3FN, pero no está en FNBC, pues existe la df numSocio \rightarrow nombreSocio y numSocio no es una clave candidata. De la misma manera se puede razonar con la df nombreSocio \rightarrow numSocio



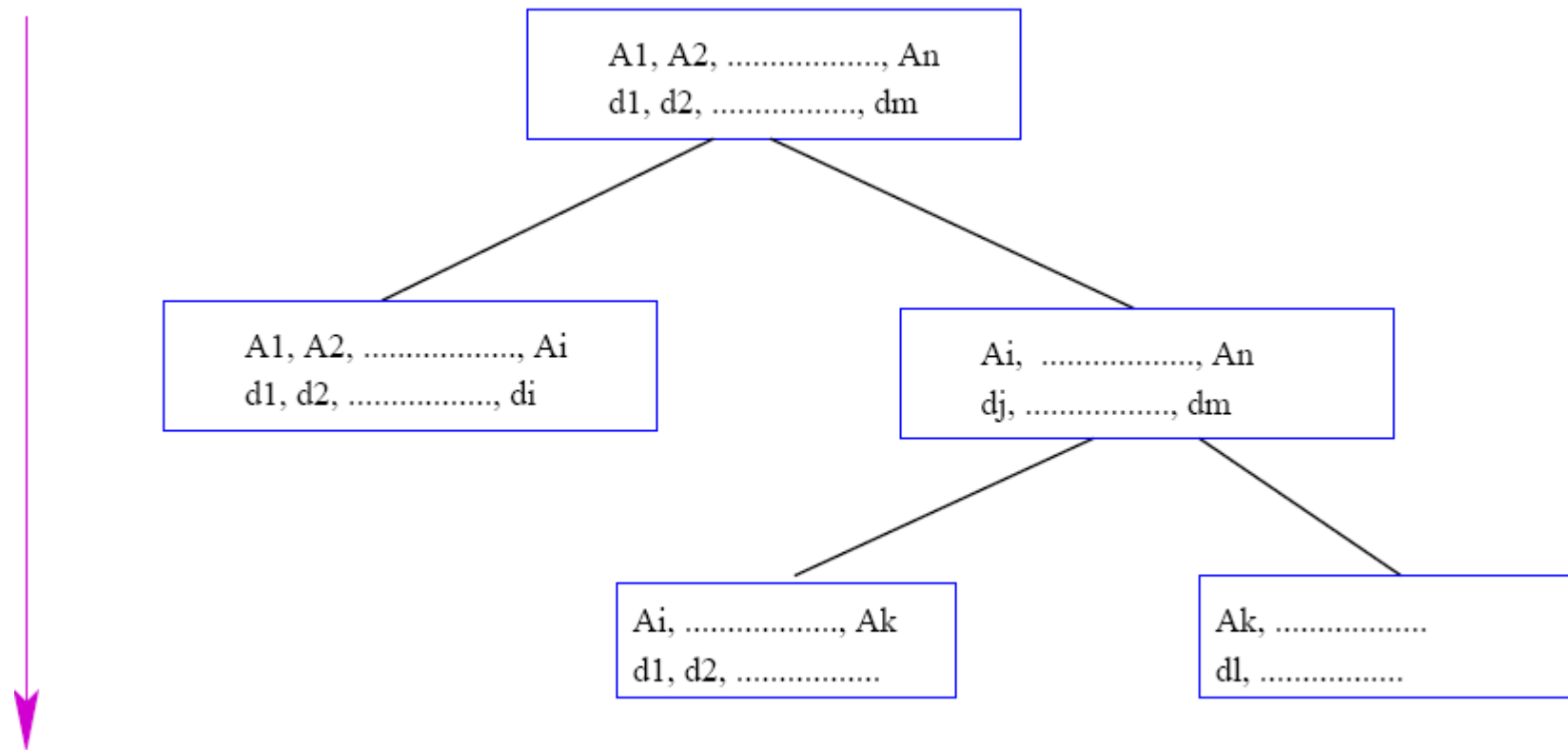
Dependencias Funcionales

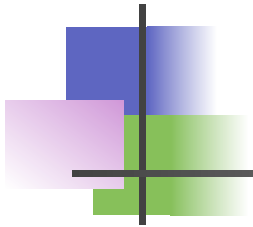
- **Forma Normal de Boyce y Codd (FNBC):**
 - La FNBC **no se cumple** cuando:
 - Existe algún determinante que no es clave candidata.
 - **Siempre están** en FNBC las relaciones:
 - Binarias.
 - Que están en 3FN y sus claves no se solapan.
- La existencia de claves candidatas solapadas no siempre supone que la relación no esté en FNBC.
- Puede ocurrir que ciertas relaciones que se encuentran en FNBC presenten todavía redundancias y anomalías, pero éstas ya no se deben a las DF, y por tanto, para evitarlas hay que recurrir a las formas normales avanzadas basadas en otro tipo de dependencias.



Dependencias Funcionales

■ Diseño relacional (Análisis o descomposición) :

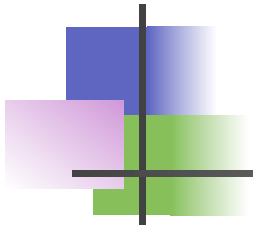




Dependencias Funcionales

■ **Análisis o descomposición :**

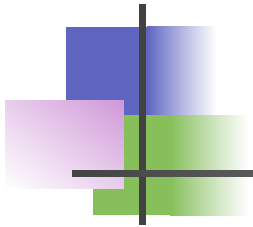
- Permite analizar una estructura relacional existente, determinando su nivel de normalización y descomponiéndola en nuevas estructuras relacionales más regulares que cumplan ciertas propiedades.
- Se trata de verificar determinados test a un esquema de relación para comprobar si se encuentra en una determinada forma normal y, en caso negativo, ir transformando en pasos sucesivos dichos esquemas en otros de menor grado, utilizando para ello el operador de proyección. Se pasa así a obtener esquemas en formas normales más avanzadas que el esquema de partida.
- La normalización, según este método, consiste en descomponer un conjunto de esquemas de relaciones que presentan anomalías y redundancia en otras de menor grado (menos atributos), intentando conservar siempre la información y las dfs.



Dependencias Funcionales

■ Descomposición en proyectos independientes:

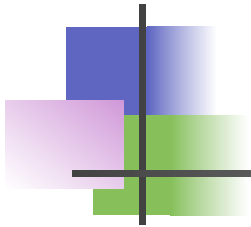
- Sea $R(A, DF)$ una relación y $R_1(A_1, DF_1)$ y $R_2(A_2, DF_2)$ dos de sus proyecciones, se dice que dichas proyecciones son independientes si, y solo si,
 1. Sus atributos comunes son la clave primaria de al menos, una relación, es decir, se debe cumplir que:
 - (a) $R_1 \cap R_2 \rightarrow R_1$
 - (b) $R_1 \cap R_2 \rightarrow R_2$
 2. Cada df en R puede deducirse de las de R_1 y R_2 , es decir, $DF^+ = (DF_1 \cap DF_2)^+$



Dependencias Funcionales

■ Ejemplo:

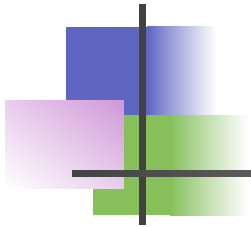
- Dada la relación LIBRO(codLibro, editorial, pais) y las dependencias funcionales $DF = \{\text{codLibro} \rightarrow \text{editorial}, \text{editorial} \rightarrow \text{pais}\}$ que está en 2FN, pero no en 3FN, existen 3 posibilidades de descomposición.
 1. En la cual se obtienen las siguientes relaciones
LIBRO1(codLibro, pais) con $DF_1 = \{\text{codLibro} \rightarrow \text{pais}\}$ y
LIBRO2(editorial, pais) con $DF_2 = \{\text{editorial} \rightarrow \text{pais}\}$
 2. Que daría como resultado
LIBRO3(codLibro, editorial) con $DF_1 = \{\text{codLibro} \rightarrow \text{editorial}\}$ y
LIBRO4(codLibro, pais) con $DF_2 = \{\text{codLibro} \rightarrow \text{pais}\}$
 3. Que logra las siguientes relaciones
LIBRO5(codLibro, editorial) con $DF_1 = \{\text{codLibro} \rightarrow \text{editorial}\}$ y
LIBRO6(editorial, pais) con $DF_2 = \{\text{editorial} \rightarrow \text{pais}\}$
- Analizar cada una de las descomposiciones y determinar las que son sin pérdida de información y sin pérdida de dfs (proyecciones independientes).



Dependencias Funcionales

■ Ejemplo:

- Dada la relación LIBRO(codLibro, editorial, pais) y las dependencias funcionales $DF = \{\text{codLibro} \rightarrow \text{editorial}, \text{editorial} \rightarrow \text{pais}\}$ que está en 2FN, pero no en 3FN, existen 3 formas de descomposición.
 1. En la cual se obtienen las siguientes relaciones
LIBRO1(codLibro, pais) con $DF_1 = \{\text{codLibro} \rightarrow \text{pais}\}$ y
LIBRO2(editorial, pais) con $DF_2 = \{\text{editorial} \rightarrow \text{pais}\}$
 2. Que daría como resultado
LIBRO3(codLibro, editorial) con $DF_1 = \{\text{codLibro} \rightarrow \text{editorial}\}$ y
LIBRO4(codLibro, pais) con $DF_2 = \{\text{codLibro} \rightarrow \text{pais}\}$
 3. Que logra las siguientes relaciones
LIBRO5(codLibro, editorial) con $DF_1 = \{\text{codLibro} \rightarrow \text{editorial}\}$ y
LIBRO6(editorial, pais) con $DF_2 = \{\text{editorial} \rightarrow \text{pais}\}$
- Sin pérdida de información: Atributo común editorial, clave de una de las relaciones.
- Sin pérdida de dfs: Todas las dependencias de Libro se conservan, por transitividad $\text{codLibro} \rightarrow \text{pais}$



Dependencias Funcionales

■ Descomposición en FNBC:

- Para las tres primeras formas normales, siempre es posible conseguir proyecciones independientes,
- es decir, es posible conseguir una descomposición sin pérdida de información y sin pérdida de dfs.
- Sin embargo, a veces no es posible alcanzar que una relación alcance la FNBC sin pérdida de información o dfs.



Dependencias Funcionales

■ Descomposición en FNBC:

Ejemplo 1:

- Sea la relación PROFESORES(rutP, nombreP, rutA, calif) con las dfs
$$DF = \{rutP \leftrightarrow nombreP, (rutP, rutA) \rightarrow calif\}^a.$$
- La descomposición de esta relación se puede hacer de la siguiente manera:
PROFESOR1(rutP, nombreP) con $DF1 = \{rutP \leftrightarrow nombre\}$ y
NOTA(rutP, rutA, calif) con $DF2 = \{(rutP, rutA) \rightarrow calif\}^b$

^a Demostrar que la relación *PROFESORES* está en 2FN y 3FN pero no en FNBC.

^b Demostrar que esta descomposición es sin pérdida de información y sin pérdida de dfs y que *PROFESOR1* y *NOTA* están en FNBC.



Dependencias Funcionales

■ Descomposición en FNBC (cont):

■ **Ejemplo 2:** Sea la relación

LIBRO(titulo, coleccion, editorial) y las dfs

$DF = \{(titulo, editorial) \rightarrow coleccion, coleccion \rightarrow editorial\}^a$.

■ Posibles descomposiciones^b

1. RES1(titulo, editorial) con $DF1 = \emptyset$

RES2(editorial, coleccion) con $DF2 = \{coleccion \rightarrow editorial\}$

2. RES3(titulo, coleccion) con $DF1 = \emptyset$

RES4(titulo, editorial) con $DF2 = \emptyset$

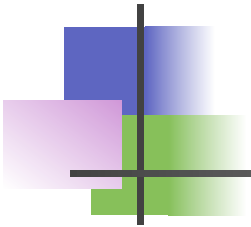
3. RES5(titulo, coleccion) con $DF1 = \emptyset$

RES6(coleccion, editorial) con $DF2 = \{coleccion \rightarrow editorial\}$

■ No es posible descomponer la relación *LIBRO* en dos relaciones independientes. Se debe decidir si *LIBRO* queda en 3FN o se pasa a FNBC con pérdida de información o pérdida de dfs.

■ ^a Demostrar que la relación *LIBRO* está en 2FN y 3FN pero no en FNBC.

■ ^b Demostrar que ninguna de las descomposiciones de la relación *LIBRO* es sin pérdida de información y sin pérdida de dfs



Dependencias Funcionales

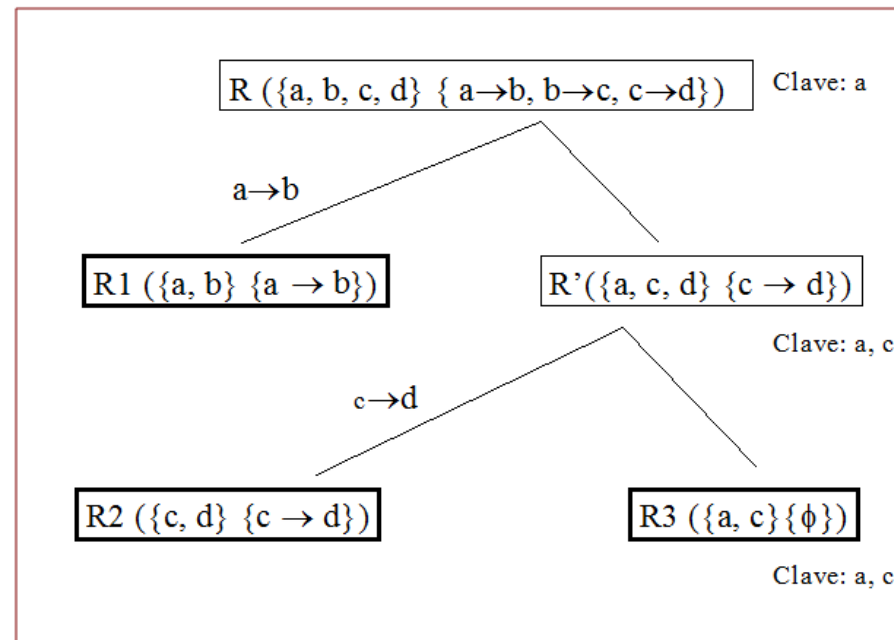
■ Proceso de Descomposición :

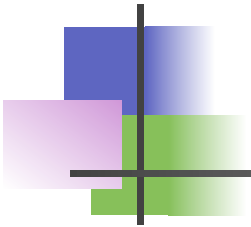
- Dado el esquema $R(A, DF)$, donde A es un conjunto de atributos y DF un conjunto de dependencias funcionales. Los pasos a seguir en un proceso de descomposición para aumentar el nivel normalización, son:
 1. **Hallar un recubrimiento minimal** DF^m .
 2. **Determinar la(s) clave(s)**, así como los **atributos principales y no principales**.
 3. **Identificar la FN** en que se encuentra la relación. Si dicha FN es la buscada acabar, en caso contrario continuar.
 4. **Agrupar las DF** que tengan el **mismo implicante**.
 5. **Obtener proyecciones independientes** sobre cada uno de los **grupos de DF**, de forma que los atributos que aparecen en el correspondiente grupo constituyen una nueva relación, y los implicados del grupo no aparecen en la relación restante.
 6. **Repetir el paso 5** hasta que no pueda continuarse porque todas las dependencias estén implicadas por una clave (en este caso, para llegar hasta FNBC hay que perder dependencias; y es decisión del diseñador parar el proceso en la 3FN o avanzar hasta FNBC con el inconveniente señalado).

Dependencias Funcionales

- **Es relevante el orden** en el que se van tomando los grupos de dependencias funcionales: los atributos que desaparecen de la relación origen deben ser aquellos que no entran a formar parte de ninguna otra dependencia (en caso contrario se perderían dependencias).

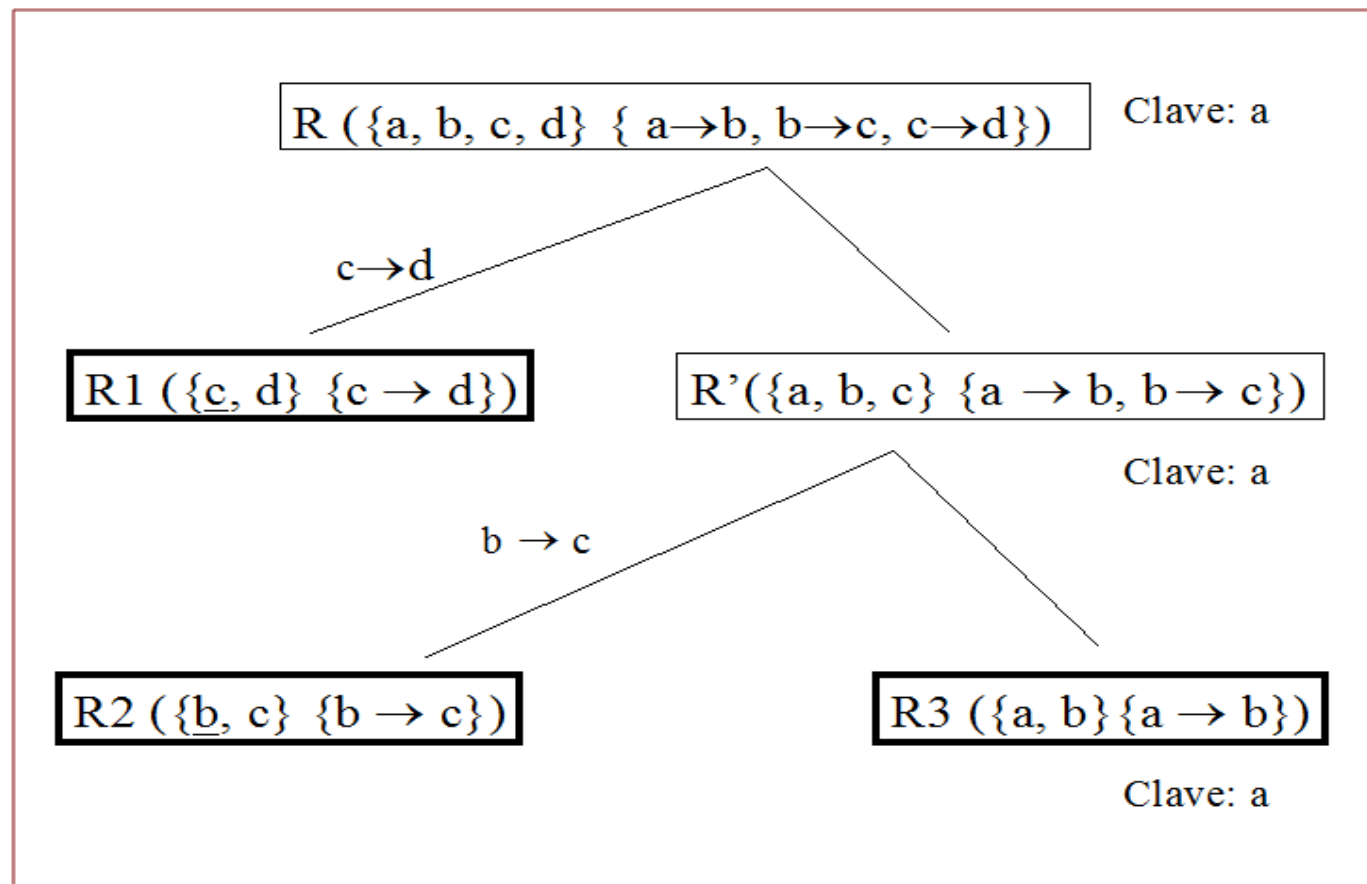
Incorrecto:
se pierde
 $b \rightarrow c$

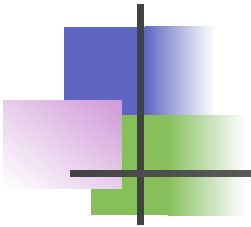




Dependencias Funcionales

Correcto: se debe comenzar separando $c \rightarrow d$





Normalización

- Introducción al refinamiento de esquemas
- Dependencias Funcionales (DFs)
- Razonamiento sobre DFs
- ➡ ■ Dependencias Multivaluadas, de Proyección y Unión

Dependencias Multivaluadas, de Proyección y Unión

■ Dependencias Multivaluadas y Cuarta Forma Normal:

AUTORES (No está en 1FN)

AUTOR	MATERIA	INSTITUCION
Date	Lenguaje SQL	Relational Inst.
	Diseño BD	Codd & Date Consultores
Ullman	Diseño BD	Stanford Univ.
	Bases Conoc.	

$DF = \emptyset$

AUTORES1 (Está en 1FN, 2FN, 3FN y FNBC)^a

AUTOR	MATERIA	INSTITUCION
Date	Lenguaje SQL	Relational Inst.
Date	Lenguaje SQL	Codd & Date Consultores
Date	Diseño BD	Relational Inst.
Date	Diseño BD	Codd & Date Consultores
Ullman	Diseño BD	Stanford Univ.
Ullman	Bases Conoc.	Stanford Univ.

^a Demostrar que está en FNBC



Dependencias Multivaluadas, de Proyección y Unión

■ Dependencia Multivaluada (dm):

- Una dependencia multivaluada es una sentencia de la forma

$$X \twoheadrightarrow Y$$

- que se lee X multidetermina a Y , donde X e Y son descriptores tales que un cierto valor de X implica un conjunto bien definido de valores de Y , con independencia de los demás atributos de la relación.
- Dada la relación R(A), se cumple que
- si para cada valor de X hay un conjunto de valores de cero o más valores de Y, independiente de los valores de los atributos A – X – Y. Por ejemplo, en la tabla AUTORES1 se cumplen las siguientes dependencias multivaluadas:

$$\text{AUTOR} \twoheadrightarrow \text{MATERIA}$$

$$\text{AUTOR} \twoheadrightarrow \text{INSTITUCION}$$



Dependencias Multivaluadas, de Proyección y Unión

■ Definición Dependencia Multivaluada:

- Dado un esquema $R(A)$ la $dm\ X \twoheadrightarrow Y$ se verificará en R si, y sólo si, para toda tupla u y v de cualquier extensión r de R tales que:
 1. $u[X] = v[X]^a$
 2. $u[Y] \neq v[Y]$
 3. $u[A - X - Y] \neq v[A - X - Y]$
 - existen tuplas t y w de r tales que:
 1. $u[X] = v[X] = t[X] = w[X]$
 2. $t[Y] = v[Y]$ y $t[A - X - Y] = u[A - X - Y]$
 3. $w[Y] = u[Y]$ y $w[A - X - Y] = v[A - X - Y]$
-
- ^a la notación $u[X]$, $v[X]$, etc. significa la proyección de la tupla u , v , etc. sobre los atributos, en este caso, X



Dependencias Multivaluadas, de Proyección y Unión

■ Otra Definición de Dependencia Multivaluada:

- Dado el esquema $R(A)$, donde X e Y son descriptores y $Z = A - (X \cup Y)$, existe la dm $X \twoheadrightarrow Y$, si para cualquier valor (x, z) de los atributos X y Z se cumple que

$$R[x, Y, z] = R[x, Y]$$

- La notación anterior significa que los valores de Y asociados a (x, z) son iguales a los de Y asociados a x . En esta definición X , Y y Z son descriptores disjuntos.
- Las dms siempre se producen por parejas; así si en el esquema $R(A)$ existe la dm $X \twoheadrightarrow Y$, al mismo tiempo habrá de cumplirse la dm $X \twoheadrightarrow A - (X \cup Y)$ lo que se representa por

$$X \twoheadrightarrow Y \mid A - (X \cup Y)$$

- En el ejemplo de AUTORES1

$$\text{autor} \twoheadrightarrow \text{materia} \mid \text{institucion}$$



Dependencias Multivaluadas, de Proyección y Unión

■ Ejemplo donde una dm no se cumple:

- A diferencia de las dfs donde el que se verifique que $X \rightarrow Y$ depende exclusivamente de los descriptores, en las dms influye el resto de los atributos de la relación como lo demuestra el siguiente ejemplo. Suponiendo que la relación AUTORES anterior se le agrega el atributo *departamento* que indica “**El departamento de una institución en el que se trabaja en una cierta materia**”

AUTOR-NUEVA

AUTOR	MATERIA	INSTITUCION	DEPARTAMENTO
Date	Lenguaje SQL	Relational Inst.	Lenguajes
Date	Lenguaje SQL	Codd & Date Cons.	Bases de datos
Date	Diseño BD	Relational Inst.	Bases de datos
Date	Diseño BD	Codd & Date Cons.	Análisis
Ullman	Diseño BD	Stanford Univ.	Lenguajes
Ullman	Bases Conoc.	Stanford Univ.	Inteligencia Artificial



Dependencias Multivaluadas, de Proyección y Unión

■ Ejemplo donde una dm no se cumple (cont.):

AUTOR-NUEVA	AUTOR	MATERIA	INSTITUCION	DEPARTAMENTO
	Date	Lenguaje SQL	Relational Inst.	Lenguajes
	Date	Lenguaje SQL	Codd & Date Cons.	Bases de datos
	Date	Diseño BD	Relational Inst.	Bases de datos
	Date	Diseño BD	Codd & Date Cons.	Análisis
	Ullman	Diseño BD	Stanford Univ.	Lenguajes
	Ullman	Bases Conoc.	Stanford Univ.	Inteligencia Artificial

- AUTOR → MATERIA no se cumple pues
AUTOR-NUEVA["Date",materia] = { "Lenguaje SQL", "Diseño BD" }
- Mientras que
AUTOR-NUEVA["Date",materia, "Lenguajes","Relational Inst."]=
{ "Lenguaje SQL" }



Dependencias Multivaluadas, de Proyección y Unión

■ Dependencias Multivaluadas Triviales:

- Las dms serán triviales en los casos en que:

1. $X = Y$
2. $Y \subset X$
3. $X \cup Y = A$



Dependencias Multivaluadas, de Proyección y Unión

■ Cuarta Forma Normal (4FN):

- Diremos que una relación se encuentra en 4FN, si, y sólo si las únicas dms no triviales son aquellas en las cuales una clave multidetermina un atributo, es decir, toda dm viene determinada por una clave candidata.



Dependencias Multivaluadas, de Proyección y Unión

■ Cuarta Forma Normal (4FN):

- En los ejemplos que hemos visto de AUTORES(autor, materia, institución) donde existen las dms
autor → materia, y
autor → institución
- la relación AUTORES no está en 4FN, ya que estas dms están implicadas por *autor* que no es una clave candidata (la clave candidata es el descriptor (autor, materia, institución)) Para evitar las anomalías que se pueden producir al actualizar esta relación que no está en 4FN, es preciso descomponerla en dos proyecciones, quedando:
AUTORES1(autor, materia) y
AUTORES2(autor, institución)
- que sí están en 4FN.



Dependencias Multivaluadas, de Proyección y Unión

- **Descomposición sin pérdida de información:**

- Dado el esquema de relación $R(A, DEP)$ donde DEP es un conjunto de dependencias (funcionales y multivaluadas) las proyecciones R_1 y R_2 cumplen la propiedad de descomposición sin pérdida de información si se cumple :

1. $R_1 \cap R_2 \twoheadrightarrow R_1 - R_2$

2. $R_1 \cap R_2 \twoheadrightarrow R_2 - R_1$



Dependencias Multivaluadas, de Proyección y Unión

- **Dependencias de reunión y Quinta Forma Normal (5FN):**

EDITA

EDITORIAL	IDIOMA	TEMA
Addison Wesley	inglés	Base de datos
Addison Wesley	español	CASE
Prentice Hall	español	Base de datos
Addison Wesley	español	Base de datos

- EDITA indica que una editorial publica en un determinado idioma sobre ciertos temas. Los atributos de EDITA no son independientes, existiendo entre ellos ciertas restricciones.



Dependencias Multivaluadas, de Proyección y Unión

■ Dependencias de reunión y (5FN):

EDITA

EDITORIAL	IDIOMA	TEMA
Addison Wesley	inglés	Base de datos
Addison Wesley	español	CASE
Prentice Hall	español	Base de datos
Addison Wesley	español	Base de datos

- Las restricciones se pueden expresar como:
"Si una editorial (Addison-Wesley) publica sobre un determinado tema (Bases de Datos); si, además, esa editorial (Addison-Wesley) publica en un cierto idioma (español), y si se está publicando (en este caso Prentice Hall) sobre ese tema (Bases de Datos) en el idioma (español), forzosamente ocurre que la editorial (Addison-Wesley) publica sobre el tema (Bases de Datos) en el idioma español"



Dependencias Multivaluadas, de Proyección y Unión

■ Dependencias de reunión y (5FN):

EDITA

EDITORIAL	IDIOMA	TEMA
Addison Wesley	inglés	Base de datos
Addison Wesley	español	CASE
Prentice Hall	español	Base de datos
Addison Wesley	español	Base de datos

- Si sólo existieran las dos primeras tuplas e insertamos la tercera, también debemos insertar la última tupla, es decir, <Adisson Wesley, español, Bases de Datos>



Dependencias Multivaluadas, de Proyección y Unión

■ Descomposición en tres proyecciones de EDITA

EDITA1

EDITORIAL	IDIOMA
Addison Wesley	Inglés
Addison Wesley	Español
Prentice Hall	Español
Adisson Wesley	Español

EDITA2

IDIOMA	TEMA
Inglés	Base de Datos
Español	CASE
Español	Bases de Datos
Español	Bases de Datos

EDITA3

EDITORIAL	TEMA
Addison Wesley	Bases de Datos
Addison Wesley	CASE
Prentice Hall	Bases de Datos
Addison Wesley	Bases de Datos

$$\text{EDITA} = \text{EDITA1} * \text{EDITA2} * \text{EDITA3}$$



Dependencias Multivaluadas, de Proyección y Unión

■ Definición formal de dependencia de reunión (DR)

- Una dependencia de Reunión DR denotada por $DR(R_1, R_2, \dots, R_n)$ especificada sobre el esquema de relación R, especifica una restricción sobre los estados r de R. La restricción establece que todo estado permitido r de R debe tener una descomposición de reunión sin pérdida de información para dar R_1, R_2, \dots, R_n ; esto es:

$$*(\pi_{R_1}(r), \pi_{R_2}(r), \dots, \pi_{R_n}(r)) = r$$

■ Observación:

Si R presenta la propiedad de DR respecto de R_1, R_2 y R_3 , entonces $\langle A_1, B_1, C_1 \rangle \in R$ si, y sólo si, $\langle A_1, B_1 \rangle \in R_1$, $\langle B_1, C_1 \rangle \in R_2$ y $\langle C_1, A_1 \rangle \in R_3$. Es decir, que no pueden aparecer en R solas las tuplas $\langle A_1, B_1, C_2 \rangle$, $\langle A_2, B_1, C_1 \rangle$ y $\langle A_1, B_2, C_1 \rangle$, si no que debe aparecer junto a ellas la tupla $\langle A_1, B_1, C_1 \rangle$

- Una DR $DR(R_1, R_2, \dots, R_n)$ especificada sobre el esquema R es una **DR Trivial** si uno de los esquemas de relación R_i es igual a R.



Dependencias Multivaluadas, de Proyección y Unión

■ Quinta Forma Normal (5FN)

- Un Esquema R está en 5FN respecto de un conjunto D de dfs, dms y DR si, para cada dependencia de reunión no trivial $DR(R_1, R_2, \dots, R_n) \in D^+$ toda R_i es una clave candidata de R.
- Una relación que no se encuentre en 5FN por tener una DR sobre sus descriptores X_1, X_2, \dots, X_n puede ser descompuesta sin pérdida de información en n esquemas de relación independientes, es decir:

$$R_1(X_1), R_2(X_2), \dots, R_n(X_n)$$



Dependencias Multivaluadas, de Proyección y Unión

■ Dependencias de Inclusión

- Se definieron para formalizar los dos tipos de restricciones interrelaciones:
 - De clave foránea (no puede especificarse por dependencias funcionales o multivaluadas por que los atributos están en relaciones diferentes)
 - Restricciones entre relaciones que presentan una relación de clase/subclase
- Una dependencia de inclusión $R.X < S.Y$ entre dos conjuntos de atributos (X del esquema R e Y del esquema S) especifica la restricción de que, en cualquier momento en que r es un estado de R y s lo es de S, se debe tener que

$$\pi_X(r(R)) \subseteq \pi_Y(s(S))$$

- Se debe cumplir que $X = \{A_1, A_2, \dots, A_n\}$ y $Y = \{B_1, B_2, \dots, B_n\}$, $\text{Dom}(A_i)$ debe ser compatible con $\text{Dom}(B_i)$, para $1 \leq i \leq n$



Dependencias Multivaluadas, de Proyección y Unión

■ Dependencias de Inclusión

■ Ejemplos:

- DEPARTAMENTO.NSS < EMPLEADO.NSS (clave foránea)
- TRABAJA_EN.NSS < EMPLEADO.NSS (clave foránea)
- EMPLEADO.NSS < PERSONA.NSS (Clase/subclase)
- ALUMNO.NSS < PERSONA.NSS (Clase/subclase)

■ Estas dependencias también cuentan con reglas de inferencia siendo tres de ellas:

- Reflexividad: $R.X < R.X$
- Correspondencia de atributo: si $R.X < S.Y$, donde $X = \{A_1, A_2, \dots, A_n\}$ e $Y = \{B_1, B_2, \dots, B_n\}$ y A_i se corresponde con B_i , entonces $R.A_i < S.B_i$, para $1 \leq i \leq n$.
- Transitividad: Si $R.X < S.Y$ y $S.Y < T.Z$, entonces $R.X < T.Z$

■ Hasta ahora no se han desarrollado formas normales basadas en este tipo de dependencias



Dependencias Multivaluadas, de Proyección y Unión

- **Dependencias df's basadas en funciones aritméticas y procedimientos**

- A veces, los atributos de una relación pueden estar relacionados por una función o procedimiento.
- **Ejemplo:**

LINEA_PEDIDO(númeroPedido, NúmeroObjeto, Cantidad, PrecioUnitario, PrecioTotal, PrecioDescuento)

- *Podemos tener las siguientes df's*
(cantidad, PrecioUnitario) → PrecioTotal
- considerando $\text{PrecioTotal} = \text{PrecioUnitario} * \text{Cantidad}$
(númeroObjeto, Cantidad, PrecioUnitario) → PrecioDescuento
(númeroObjeto, Cantidad, PrecioTotal) → PrecioDescuento



Dependencias Multivaluadas, de Proyección y Unión

■ Forma Normal de Dominio-Clave

- Está basada en tres nociones:
 - **Declaración de dominio.** Sea A un atributo, y sea Dom un conjunto de valores. La declaración de dominio $A \subseteq Dom$ requiere que el valor de A de todas las tuplas sean valores en Dom .
 - **Declaración de Clave.** Todas las declaraciones de clave son df 's
 - **Restricción General.** Predicado en el conjunto de todas las relaciones de un esquema dado. Ejemplo, todas las cuentas cuyo primer dígito del número de cuenta es 9, deben tener un saldo mínimo de 3.000.000.
- El propósito de un diseño de base de datos en forma normal de dominio-clave es lograr que las restricciones generales se prueben usando solamente las restricciones de dominio y de clave.



Dependencias Multivaluadas, de Proyección y Unión

■ Forma Normal de Dominio-Clave

- Formalmente:
 - Sea D un conjunto de restricciones de dominio y K un conjunto de restricciones de clave para un conjunto de esquemas R , G representa las restricciones generales de R . El esquema R está en forma normal de Dominio-Clave si $D \cup K$ implica lógicamente G .