

Normalización - 2da. Parte

07/Abril/2017



Normalización - 2da. Parte

Normalización - Marco General

- **Normalización 1era. Parte**
 - Concepto DF
 - Problemas de DF y cómo eliminarlos por medio del método de descomposición
 - 1FN, 2FN, 3FN, BCNF
- **Normalización 2da. Parte**
 - Inferencia de DF
 - Conceptos nuevos: clausura, equivalencia y cubrimiento mínimo
 - Propiedades de la descomposición
 - Algoritmos para el diseño de esquemas

Normalización - 2da. Parte

Normalización - Reglas de Inferencia

- Diseñador de BD especifica DF semánticamente obvias
- Existencia de DF no especificadas que pueden ser inferidas
- **Ejemplo.**
 - $R = \{E_CUIL, Nro_Depto, D_Nombre\}$
 - $F = \{E_CUIL \rightarrow Nro_Depto, Nro_Depto \rightarrow D_Nombre\}$
 - De ambas DFs se puede inferir que $E_CUIL \rightarrow D_Nombre$
- **Inferencia.** Una DF $X \rightarrow Y$ es **inferida de** o **implicada por** un conjunto de DFs F de R si se cumple $X \rightarrow Y$ en toda instancia legal $r(R)$. Es decir, siempre que $r(R)$ satisface F , se cumple $X \rightarrow Y$
- **Clausura.** Conjunto de todas las DFs de F más todas las DFs que puedan ser inferidas de F . Se denota como F^+
 - $R = \{E_CUIL, Nro_Depto, D_Nombre\}$
 - $F = \{E_CUIL \rightarrow Nro_Depto, Nro_Depto \rightarrow D_Nombre\}$
 - $F^+ = \{E_CUIL \rightarrow Nro_Depto, Nro_Depto \rightarrow D_Nombre, E_CUIL \rightarrow D_Nombre, \dots\}$
- **Necesidad.** Para calcular F^+ es necesario un método: Reglas de inferencia

Normalización - 2da. Parte

Normalización - Reglas de Inferencia (Cont.)

- **Reglas de Inferencia.** Propuestas por Armstrong (1974) y conocidas como “Axiomas de Armstrong”
 - **RI1 (regla reflexiva).** Si $Y \subseteq X$, entonces $X \rightarrow Y$
 - **RI2 (regla de incremento).** $\{X \rightarrow Y\} \models XZ \rightarrow YZ$
 - **RI3 (regla transitiva).** $\{X \rightarrow Y, Y \rightarrow Z\} \models X \rightarrow Z$
- **Demostración RI1.** Supuestos
 - $Y \subseteq X$
 - t_1, t_2 existen en una instancia $r(R)$ tal que $t_1[X] = t_2[X]$Entonces, $t_1[Y] = t_2[Y]$ dado que $Y \subseteq X$; por lo tanto $X \rightarrow Y$ en r .
- **Demostración RI2. (por contradicción)** Supuestos
 - $X \rightarrow Y$ se cumple en $r(R)$
 - $XZ \rightarrow YZ$ NO se cumple en $r(R)$Entonces existen t_1, t_2 tal que
 - 1 $t_1[X] = t_2[X]$
 - 2 $t_1[Y] = t_2[Y]$
 - 3 $t_1[XZ] = t_2[XZ]$
 - 4 $t_1[YZ] \neq t_2[YZ]$Esto no es posible dado que de (1) y (3) se deduce (5) $t_1[Z] = t_2[Z]$, y de (2) y (5) se obtiene (6) $t_1[YZ] = t_2[YZ]$, contradiciendo (4)

Normalización - 2da. Parte

Normalización - Reglas de Inferencia (Cont.)

• Demostración RI3. Supuestos

- 1 $X \rightarrow Y$ se cumple en $r(R)$
- 2 $Y \rightarrow Z$ se cumple en $r(R)$

Entonces para cualquier t_1, t_2 en $r(R)$ tal que $t_1[X] = t_2[X]$, debe pasar que (3) $t_1[Y] = t_2[Y]$ por asunción (1). También se sabe, por (3) y por asunción (2) que $X \rightarrow Z$. Por lo tanto, RI3 se cumple en $r(R)$.

• Propiedades.

- **Fiable (Sound).** Dado F de R , cualquier DF deducida de F utilizando RI1 a RI3, se cumple en cualquier estado $r(R)$ que satisface F
- **Completa (Complete).** F^+ puede ser determinado a partir de F aplicando solamente RI1 a RI3

• Reglas de Inferencia Adicionales. (corolarios de Armstrong)

- **RI4 (regla de descomposición o proyección).** $\{X \rightarrow YZ\} \models X \rightarrow Y$
- **RI5 (regla de unión o aditiva).** $\{X \rightarrow Y, X \rightarrow Z\} \models X \rightarrow YZ$
- **RI6 (regla pseudotransitiva).** $\{X \rightarrow Y, WY \rightarrow Z\} \models WX \rightarrow Z$

Normalización - Reglas de Inferencia (Cont.)

• Demostración RI4.

- 1 $X \rightarrow YZ$ (hipótesis)
- 2 $YZ \rightarrow Y$ (usando RI1 y tomando que $Y \subseteq YZ$)
- 3 $X \rightarrow Y$ (usando RI3 sobre (1) y (2))

• Demostración RI5.

- 1 $X \rightarrow Y$ (hipótesis)
- 2 $X \rightarrow Z$ (hipótesis)
- 3 $X \rightarrow XY$ (usando RI2 sobre (1) incrementando con X ; notar que $XX = X$)
- 4 $XY \rightarrow YZ$ (usando RI2 sobre (2) incrementando con Y)
- 5 $X \rightarrow YZ$ (usando RI3 sobre (3) y (4))

• Demostración RI6.

- 1 $X \rightarrow Y$ (hipótesis)
- 2 $WY \rightarrow Z$ (hipótesis)
- 3 $WX \rightarrow WY$ (usando RI2 sobre (1) incrementando con W)
- 4 $WX \rightarrow Z$ (usando RI3 sobre (3) y (2))

• Decidir si es verdadero o falso

- $X \rightarrow A$ y $Y \rightarrow B$, entonces $XY \rightarrow AB$ **verdadero**
- $XY \rightarrow A$, entonces $X \rightarrow A$ o $Y \rightarrow A$ **falso** (¿ejemplo?)

Normalización - Clausura

• Diseño. Típicamente

- 1 Diseñador especifica conjunto de DFs F determinadas por semántica de atributos de R
- 2 Se utilizan RI1 a RI3 para inferir DFs adicionales

• Punto (2) de manera sistemática

- determinar conjunto de atributos X que aparecen del lado izq. de DFs de F
- determinar conjunto Y de todos los atributos que dependen de X

• Clausura de X . Conjunto de atributos que son determinados por X basados en F . Se nota X^+

• Algoritmo Nro. 1 para determinar X^+

Entrada: DFs F de R ; subconjunto de atributos X de R

1. $X^+ := X$
2. **repetir**
3. $viejoX^+ := X^+$
4. **Para cada** $DF Y \rightarrow Z$ **en** F **hacer**
5. **Si** $Y \subseteq X^+$ **entonces** $X^+ = X^+ \cup Z$
6. **hasta** $(X^+ = viejoX^+)$

Normalización - Clausura (Cont.)

• Ejemplo.

- $R = (idClase, Curso, Instrumento, Puntos, Libro, Editor, Aula, Capacidad)$
- $F = \{$
 $DF1: idClase \rightarrow \{Curso, Instrumento, Puntos, Libro, Editor, Aula, Capacidad\},$
 $DF2: Curso \rightarrow Puntos,$
 $DF3: \{Curso, Instrumento\} \rightarrow \{Libro, Aula\},$
 $DF4: Libro \rightarrow Editor,$
 $DF5: Aula \rightarrow Capacidad$
 $\}$

• Aplicando el algoritmo para obtener X^+

- $\{idClase\}^+ = \{idClase, Curso, Instrumento, Puntos, Libro, Editor, Aula, Capacidad\} = R$
- $\{Curso\}^+ = \{Curso, Puntos\}$
- $\{Curso, Instrumento\}^+ = \{Curso, Instrumento, Puntos, Libro, Editor, Aula, Capacidad\}$

• Observación. Clausura $idClase \not\subseteq \{Curso, Instrumento\}^+$ por lo tanto NO es CK

Normalización - Equivalencia

- **Cubrimiento.** Dados E y F conjuntos de DFs, F **cubre** a E si $(\forall df \in E) df \in F^+$
- **Equivalencia.** Dados E y F conjuntos de DFs, F y E son **equivalentes** si $F^+ = E^+$, es decir, si F **cubre** a E y E **cubre** a F
- **Ejercicio.** Decir si los siguientes conjuntos de DFs son equivalentes
 - $F = \{A \rightarrow C, AC \rightarrow D, E \rightarrow AD, E \rightarrow H\}$
 - $G = \{A \rightarrow CD, E \rightarrow AH\}$
- **Procedimiento.** Para determinar si F **cubre** a G , calcular, para cada DF $X \rightarrow Y$ de G , X^+ con respecto a F . Luego verificar si este X^+ incluye los atributos en Y . Similar razonamiento para verificar si G **cubre** a F

Normalización - Conjunto Minimal de DFs

- \rightarrow Se explica cómo expandir F a F^+
- \leftarrow Se quiere ver el camino inverso, reducir F a su expresión minimal
- **Atributo Extraño.** Atributo que puede ser removido sin alterar la clausura del conjunto de DFs.
- **Formalmente.** Sea $X \rightarrow A$ en F , $Y \subset X$ es **extraño** si F implica lógicamente $(F - \{X \rightarrow A\} \cup \{Y \rightarrow A\})$
- **Características de un Conjunto de DFs para ser minimal**
 - 1 Cada DF de F debe poseer un solo atributo en su lado derecho
 - 2 No es posible reemplazar ninguna DF $X \rightarrow A$ de F por $Y \rightarrow A$, siendo $Y \subset X$, y seguir teniendo un conjunto de DFs equivalente a F
 - 3 No es posible remover ninguna DF de F y seguir teniendo un conjunto de DFs equivalente a F
- **Intuitivamente.** F **minimal** es un conjunto *canónico* y *sin redundancia*
- **Cubrimiento minimal.** Un **cubrimiento minimal** de F es un conjunto minimal de DFs (en forma canónica y sin redundancia) que es equivalente a F .
- **Existencia.** Siempre es posible hallar al menos un **cubrimiento minimal** F para cualquier conjunto de DFs E usando el siguiente algoritmo

Normalización - Conjunto Minimal de DFs (Cont.)

Algoritmo Nro. 2 Búsqueda de un cubrimiento minimal F para un conjunto de DFs E

Entrada: Conjunto de DFs E

0. $F := E$
1. **Reemplazar cada DF** $X \rightarrow \{A_1, A_2, \dots, A_n\}$ en F por n DFs $X \rightarrow A_1, X \rightarrow A_2, \dots, X \rightarrow A_n$
/*Traslada a todas las DFs a una forma canónica para los pasos subsiguientes*/
2. **Para cada DF** $X \rightarrow A$ en F
 Para cada atributo B que es un elemento de X
 Si $\{F - \{X \rightarrow A\}\} \cup \{(X - \{B\}) \rightarrow A\}$ es equivalente a F
 Entonces reemplazar $X \rightarrow A$ por $(X - \{B\}) \rightarrow A$
/*Remueve al atributo extraño B del lado izquierdo de X siempre que es posible*/
3. **Para cada DF** $X \rightarrow A$ en F
 Si $F - \{X \rightarrow A\}$ es equivalente a F
 Remover $X \rightarrow A$ de F
/*Remueve las DF redundantes siempre que es posible*/

Normalización - Conjunto Minimal de DFs (Cont.)

- **Ejemplo 1.** Sea un conjunto de DFs $E = \{B \rightarrow A, D \rightarrow A, AB \rightarrow D\}$. Encontrar el cubrimiento minimal de E denominado F
 - **Paso (1)** Todas las DFs de E están en forma canónica. No es necesario hacer ningún cambio
 - **Paso (2)** Hay que determinar si $AB \rightarrow D$ posee algún atributo extraño en su lado izquierdo. Esto es, si puede ser reemplazado por $A \rightarrow D$ o $B \rightarrow D$
 - Aplicando RI2 a $B \rightarrow A$, incrementándolo con B , se obtiene $BB \rightarrow AB$ que equivale a (i) $B \rightarrow AB$; Adicionalmente se tiene la DF (ii) $AB \rightarrow D$
 - Aplicando la RI3 (transitiva) sobre (i) y (ii), se obtiene $B \rightarrow D$. Así, $AB \rightarrow D$ puede ser reemplazada por $B \rightarrow D$
 - El conjunto original E puede ser reemplazado por otro equivalente $E' = \{B \rightarrow A, D \rightarrow A, B \rightarrow D\}$
 - No es posible otra reducción ya que todos los lados izquierdos poseen un solo atributo
 - **Paso (3)** Usando RI3 (transitiva) sobre $B \rightarrow D$ y $D \rightarrow A$, se infiere $B \rightarrow A$. Por lo tanto $B \rightarrow A$ es redundante y puede ser eliminada de E'
 - **Cubrimiento minimal de E .** $F = \{B \rightarrow D, D \rightarrow A\}$

Normalización - Conjunto Minimal de DFs (Cont.)

- **Ejemplo 2.** Sea un conjunto de DFs $E = \{A \rightarrow BCDE, CD \rightarrow E\}$. Encontrar el cubrimiento minimal de E denominado F
 - **Paso (1)** Al pasar todas las DFs de E a la forma canónica, se obtiene:
 $E = \{A \rightarrow B, A \rightarrow C, A \rightarrow D, A \rightarrow E, CD \rightarrow E\}$
 - **Paso (2)** Hay que determinar si $CD \rightarrow E$ posee algún atributo extraño en su lado izquierdo. Esto no sucede ya que las DFs $C \rightarrow E$ / $D \rightarrow E$ no pueden ser derivadas de las otras DFs
 - **Paso (3)** Vericamos si alguna DF es redundante. Dado que $A \rightarrow CD$ y $CD \rightarrow E$, por RI3 (transitiva) $A \rightarrow E$ es redundante.
 - **Cubrimiento minimal de E .** $F = \{A \rightarrow BCD, CD \rightarrow E\}$ (combinando partes derechas)

Normalización - Clave de una Relación

Algoritmo Nro. 3 Búsqueda de una clave K de R a partir de un conjunto de DFs

Entrada: Relación R y un Conjunto de DFs F de R

1. $K := R$
2. Para cada atributo $A \in K$
 Computar $(K - A)^+$ con respecto a F
 Si $(K - A)^+$ contiene todos los atributos de R entonces $K := K - \{A\}$

- Algoritmo determina una sola de las CK. Depende fuertemente de la manera en que son removidos los atributos

Normalización - Insuficiencia de formas normales

- **Descomposición.** Es la descomposición de R en un conjunto de esquemas $D = \{R_1, R_2, \dots, R_m\}$ de R

- **Propiedad deseable Nro. 1.** Se desea **preservación de atributos**

$$\bigcup_{i=1}^m R_i = R$$

Normalización - Preservación de DFs

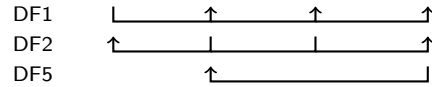
- **Propiedad deseable Nro. 2.** Si $X \rightarrow Y$ en F , es deseable que o bien aparezca en algún esquema R_i de D o bien pueda ser inferida de las DFs de algún esquema R_i
- **Importante.** No es necesario que las DFs de F aparezcan en las R_i de D . Es suficiente que la unión de las DFs de cada R_i de D sea equivalente a F
- **Proyección.** Dado un conjunto de DFs F de R , la proyección de F sobre R_i , denotado como $\pi_{R_i}(F)$ donde R_i es un subconjunto de R , es el conjunto de DFs $X \rightarrow Y$ en F^+ tal que los atributos $(X \cup Y) \subseteq R_i$
- **Preservación de DFs.** La descomposición $D = \{R_1, R_2, \dots, R_m\}$ de R **preserva dependencias** con respecto a F si la unión de las proyecciones de F de cada R_i de D es equivalente a F . Es decir, si $(\pi_{R_1}(F) \cup \dots \cup \pi_{R_m}(F))^+ = F^+$

Normalización - Preservación de DFs (Cont.)

Ejemplo 1.

LOTES_1A

id_Nacional	Provincia	id_Provincial	Zonificación
-------------	-----------	---------------	--------------



Descomposición Boyce-Codd FN (BCFN).

LOTES_1AX

id_Nacional	Zonificación	id_Provincial
-------------	--------------	---------------

LOTES_1AY

Zonificación	Provincia
--------------	-----------

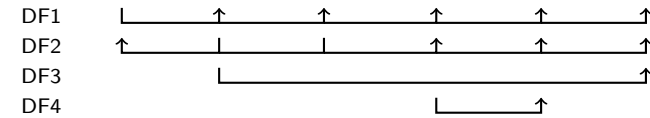
- ¿Esta descomposición preserva atributos? ¡Sí!
- ¿Esta descomposición preserva DFs? ¡NO! Se pierde DF 2

Normalización - Preservación de DFs (Cont.)

Ejemplo 2.

LOTES

id_Nacional	Provincia	id_Provincial	Zonificación	Precio_m2	Tasa_Impuesto
-------------	-----------	---------------	--------------	-----------	---------------



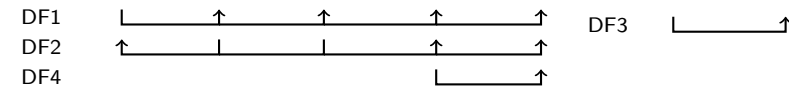
Descomposición en 2FN.

LOTES.1

id_Nacional	Provincia	id_Provincial	Zonificación	Precio_m2
-------------	-----------	---------------	--------------	-----------

LOTES.2

Provincia	Tasa_Impuesto
-----------	---------------



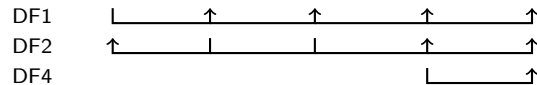
- ¿Esta descomposición preserva atributos? ¡Sí!
- ¿Esta descomposición preserva DFs? ¡Sí!

Normalización - Preservación de DFs (Cont.)

Ejemplo 3.

LOTES.1

id_Nacional	Provincia	id_Provincial	Zonificación	Precio_m2
-------------	-----------	---------------	--------------	-----------



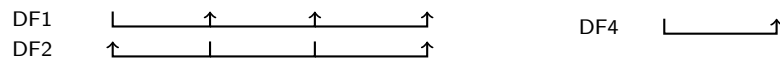
Descomposición en 3FN.

LOTES_1A

id_Nacional	Provincia	id_Provincial	Zonificación
-------------	-----------	---------------	--------------

LOTES_1B

Zonificación	Precio_m2
--------------	-----------



- ¿Esta descomposición preserva atributos? ¡Sí!
- ¿Esta descomposición preserva DFs? ¡Sí!

Afirmación Nro. 1

Siempre es posible encontrar una descomposición D con preservación de DFs con respecto a F tal que cada R_i en D se encuentre en 3FN

Normalización - Lossless Join

- Lossless Join informalmente.** El cumplimiento de esta propiedad no permite la generación de tuplas espúreas cuando se realiza un NATURAL JOIN entre las relaciones resultantes de una descomposición
- Lossless Join formalmente.** Una descomposición $D=\{R_1, R_2, \dots, R_m\}$ de R posee la propiedad **lossless join** con respecto al conjunto de DFs F de R si, para todo estado $r(R)$ que satisface F , se cumple que $\bowtie(\pi_{R_1}(r), \dots, \pi_{R_m}(r))=r$

Normalización - Lossless Join (Cont.)

Algoritmo Nro. 4 Chequeo de propiedad Lossless Join

Entrada: R , descomposición $D=\{R_1, R_2, \dots, R_m\}$ de R y un conjunto de DFs F

1. Crear una matriz S con una fila i por cada R_i en D , y una columna j por cada atributo A_j en R
2. Para todo i, j asignar $S(i, j) = b_{ij}$ /*cada b_{ij} es un elemento distinto de la matriz*/
3. Para cada i, j
Si $A_j \in R_i$ entonces $S(i, j) = a_j$ /*distingue a elementos que pertenecen a la relación R_i */
4. Repetir hasta que un loop completo no genere cambios en S
Para cada $X \rightarrow Y$ en F
Para todas las filas fs en S que tienen los mismos valores en los atributos de X
Hacer que los atributos en fs para cada columna Y tengan el mismo valor de la siguiente manera
Si alguna de las fs en Y tiene un símbolo a entonces asignarlo al resto de las fs en Y
Sino elegir arbitrariamente un símbolo b de fs en Y y asignarlo al resto de las fs en Y
5. Si alguna fila de S posee la totalidad de elementos a entonces es lossless join, caso contrario no lo es

Normalización - Lossless Join (Cont.)

Ejemplo 1. Sean

- $R = \{E_CUIL, E_Nombre, P_Número, P_Nombre, P_Ubicación, Horas\}$
- $R_1 = EMP_UBICACION = \{E_Nombre, P_Ubicación\}$
- $R_2 = EMP_PROY1 = \{E_CUIL, P_Número, P_Nombre, P_Ubicación, Horas\}$
- $D = \{R_1, R_2\}$
- $F = \{$
 $E_CUIL \rightarrow E_Nombre;$
 $P_Número \rightarrow \{P_Nombre, P_Ubicación\};$
 $\{E_CUIL, P_Número\} \rightarrow Horas;\}$

Paso 1.

	E_CUIL	E.Nombre	P.Número	P.Nombre	P.Ubicación	Horas
R_1						
R_2						

Paso 2.

	E_CUIL	E.Nombre	P.Número	P.Nombre	P.Ubicación	Horas
R_1	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}	b_{16}
R_2	b_{21}	b_{22}	b_{23}	b_{24}	b_{25}	b_{26}

Paso 3.

	E_CUIL	E.Nombre	P.Número	P.Nombre	P.Ubicación	Horas
R_1	b_{11}	a_2	b_{13}	b_{14}	a_5	b_{16}
R_2	a_1	b_{22}	a_3	a_4	a_5	a_6

- Paso 4.** No modifica ningún símbolo b en a
- Paso 5.** No hay ninguna fila en S que posea a en la totalidad de valores, por lo tanto **la descomposición no es lossless join**

Normalización - Lossless Join (Cont.)

Ejemplo 1. Sean

- $R = \{E_CUIL, E_Nombre, P_Número, P_Nombre, P_Ubicación, Horas\}$
- $R_1 = EMP = \{E_CUIL, E_Nombre\}$
- $R_2 = PROY = \{P_Número, P_Nombre, P_Ubicación\}$
- $R_3 = TRABAJA_EN = \{E_CUIL, P_Número, Horas\}$
- $D = \{R_1, R_2, R_3\}$
- $F = \{$
 $E_CUIL \rightarrow E_Nombre;$
 $P_Número \rightarrow \{P_Nombre, P_Ubicación\};$
 $\{E_CUIL, P_Número\} \rightarrow Horas;\}$

Paso 1.

	E_CUIL	E.Nombre	P.Número	P.Nombre	P.Ubicación	Horas
R_1						
R_2						
R_3						

Paso 2.

	E_CUIL	E.Nombre	P.Número	P.Nombre	P.Ubicación	Horas
R_1	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}	b_{16}
R_2	b_{21}	b_{22}	b_{23}	b_{24}	b_{25}	b_{26}
R_3	b_{31}	b_{32}	b_{33}	b_{34}	b_{35}	b_{36}

Paso 3.

	E_CUIL	E.Nombre	P.Número	P.Nombre	P.Ubicación	Horas
R_1	a_1	a_2	b_{13}	b_{14}	b_{15}	b_{16}
R_2	b_{21}	b_{22}	a_3	a_4	a_5	b_{26}
R_3	a_1	b_{32}	a_3	b_{34}	b_{35}	a_6

Normalización - Lossless Join (Cont.)

Ejemplo 1. Sean

- $R = \{E_CUIL, E_Nombre, P_Número, P_Nombre, P_Ubicación, Horas\}$
- $R_1 = EMP = \{E_CUIL, E_Nombre\}$
- $R_2 = PROY = \{P_Número, P_Nombre, P_Ubicación\}$
- $R_3 = TRABAJA_EN = \{E_CUIL, P_Número, Horas\}$
- $D = \{R_1, R_2, R_3\}$
- $F = \{$
 $E_CUIL \rightarrow E_Nombre;$
 $P_Número \rightarrow \{P_Nombre, P_Ubicación\};$
 $\{E_CUIL, P_Número\} \rightarrow Horas;\}$

Paso 4. $E_CUIL \rightarrow E_Nombre$

	E_CUIL	E.Nombre	P.Número	P.Nombre	P.Ubicación	Horas
R_1	a_1	a_2	b_{13}	b_{14}	b_{15}	b_{16}
R_2	b_{21}	b_{22}	a_3	a_4	a_5	b_{26}
R_3	a_1	b_{32}	a_3	b_{34}	b_{35}	a_6

- Paso 4.** $P_Número \rightarrow \{P_Nombre, P_Ubicación\}$

	E_CUIL	E.Nombre	P.Número	P.Nombre	P.Ubicación	Horas
R_1	a_1	a_2	b_{13}	b_{14}	b_{15}	b_{16}
R_2	b_{21}	b_{22}	a_3	a_4	a_5	b_{26}
R_3	a_1	b_{32}	a_3	b_{34}	b_{35}	a_6

- Paso 4.** $\{E_CUIL, P_Número\} \rightarrow Horas$ no produce cambios en S
- Paso 4.** Nueva vuelta sobre TODAS las DFs F no produce cambios en S
- Paso 5.** Última fila de S posee la totalidad de sus valores en a , por lo tanto **la descomposición es lossless join**

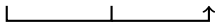

Normalización - Lossless Join para Descomposición Binaria

- **Caso especial.** Existe algoritmo más sencillo en caso de descomposición binaria
- **Limitación.** Sólo descomposición binaria
- **Chequeo Lossless Join para descomposición binaria.** También denominado NJB (Nonadditive Join Test for Binary Decompositions)
- **NJB.** Una descomposición $D=\{R_1, R_2\}$ de R cumple con la propiedad de lossless join, con respecto a un conjunto de DFs F de R sí y sólo sí
 - La DF $(R_1 \cap R_2 \rightarrow R_1 - R_2) \in F^+$, o
 - La DF $(R_1 \cap R_2 \rightarrow R_2 - R_1) \in F^+$

Normalización - Lossless Join para Descomposición Binaria

- **Ejemplo.**
CURSADA

Estudiante	Materia	Instructor
------------	---------	------------

 DF1 
 DF2 
- **Descomposición 1.** (Estudiante en ambas relaciones)

Estudiante	Instructor
------------	------------

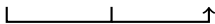

Estudiante	Materia
------------	---------

 - La DF $(R_1 \cap R_2 \rightarrow R_1 - R_2) \in F^+ \equiv (\text{Estudiante} \rightarrow \text{Instructor}) \in F^+$, o
 - La DF $(R_1 \cap R_2 \rightarrow R_2 - R_1) \in F^+ \equiv (\text{Estudiante} \rightarrow \text{Materia}) \in F^+$
- **Descomposición 1 ¿Cumple Lossless join? ¡No! porque no cumple con ninguna de las dos condiciones**

Normalización - Lossless Join para Descomposición Binaria

- **Ejemplo.**
CURSADA

Estudiante	Materia	Instructor
------------	---------	------------

 DF1 
 DF2 
- **Descomposición 2.** (Materia en ambas relaciones)

Materia	Instructor
---------	------------

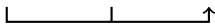

Materia	Estudiante
---------	------------

 - La DF $(R_1 \cap R_2 \rightarrow R_1 - R_2) \in F^+ \equiv (\text{Materia} \rightarrow \text{Instructor}) \in F^+$, o
 - La DF $(R_1 \cap R_2 \rightarrow R_2 - R_1) \in F^+ \equiv (\text{Materia} \rightarrow \text{Estudiante}) \in F^+$
- **Descomposición 2 ¿Cumple Lossless join? ¡No! porque no cumple con ninguna de las dos condiciones**

Normalización - Lossless Join para Descomposición Binaria

- **Ejemplo.**
CURSADA

Estudiante	Materia	Instructor
------------	---------	------------

 DF1 
 DF2 
- **Descomposición 3.** (Instructor en ambas relaciones)

Instructor	Materia
------------	---------

Instructor	Estudiante
------------	------------

 - La DF $(R_1 \cap R_2 \rightarrow R_1 - R_2) \in F^+ \equiv (\text{Instructor} \rightarrow \text{Materia}) \in F^+$, o
 - La DF $(R_1 \cap R_2 \rightarrow R_2 - R_1) \in F^+ \equiv (\text{Instructor} \rightarrow \text{Estudiante}) \in F^+$
- **Descomposición 3 ¿Cumple Lossless join? ¡Sí! porque se cumple al menos una de las dos condiciones: $(\text{Instructor} \rightarrow \text{Materia}) \in F^+$**

Normalización - Lossless Join - Descomposiciones sucesivas

- **Recapitulando.** En ejemplos previos utilizamos descomposiciones sucesivas al pasar a R a 2FN y luego a 3FN

Afirmación Nro. 2

Si se cumplen las siguientes condiciones:

- Una descomposición $D=\{R_1, R_2, \dots, R_m\}$ de R cumple la propiedad de lossless join con respecto a F de R
- Una descomposición $D_i=\{Q_1, Q_2, \dots, Q_k\}$ de R_i cumple la propiedad de lossless join con respecto a la proyección de F sobre R_i

Entonces la descomposición $D_2=\{R_1, R_2, \dots, R_{i-1}, Q_1, Q_2, \dots, Q_k, R_{i+1}, \dots, R_m\}$ de R cumple con la propiedad lossless join con respecto a F de R

Normalización - Algoritmos Diseño

- **Algoritmo D1.** Descompone relación universal R cumpliendo:
 - 3FN
 - Preservación de DFs
 - Lossless Join
- **Algoritmo D2.** Descompone relación universal R cumpliendo:
 - BCFN
 - Lossless Join
- No es posible diseñar algoritmo que produzca una descomposición en BCFN con preservación DFs y Lossless Join

Normalización - Algoritmos Diseño 1 - 3FN

- **Algoritmo Nro. D1** Descomposición en 3FN

Entrada: R universal y un conjunto de DFs F sobre R

1. Hallar el cubrimiento minimal G de F (utilizar algoritmo ya dado)
2. Para cada lado izquierdo X de cada DF que aparece en G
 Crear una relación en D con atributos $\{X \cup \{A_1\} \cup \{A_2\} \cup \dots \cup \{A_k\}\}$
 siendo $X \rightarrow A_1, X \rightarrow A_2, \dots, X \rightarrow A_k$ las únicas dependencias en G con X como lado izquierdo (X es la clave de esta relación)
3. Si ninguna relación en D contiene una clave de R
 entonces crear una relación adicional en D que contenga atributos que formen una clave de R (se puede utilizar algoritmo ya dado)
4. Eliminar relaciones redundantes de D . Una relación R de D es redundante si R es una proyección de otra relación S de D

Normalización - Algoritmos Diseño 1 - 3FN (Cont.)

- **Ejemplo 1.**
 - $U=\{E_CUIL, P_Número, E_Salario, E_Teléfono, D_Número, P_Nombre, P_Ubicación\}$
 - $F=\{ \text{FD1: } E_CUIL \rightarrow \{E_Salario, E_Teléfono, D_Número\}, \text{FD2: } P_Número \rightarrow \{P_Nombre, P_Ubicación\}, \text{FD3: } \{E_CUIL, P_Número\} \rightarrow \{E_Salario, E_Teléfono, D_Número, P_Nombre, P_Ubicación\} \}$
 - $\{E_CUIL, P_Número\}$ representa una clave de la relación U (por FD3)
 - **Paso 1.** Aplicando algoritmo de minimal cover, en su paso 3 se observa
 - $P_Número$ es atributo extraño en $\{E_CUIL, P_Número\} \rightarrow \{E_Salario, E_Teléfono, D_Número\}$
 - E_CUIL es atributo extraño en $\{E_CUIL, P_Número\} \rightarrow \{P_Nombre, P_Ubicación\}$
 - Así, cubrimiento minimal = FD1 y FD2 (FD3 es redundante). Agrupando atributos con mismo lado izq. en una sola DF:
 Cubrimiento minimal $G=\{E_CUIL \rightarrow \{E_Salario, E_Teléfono, D_Número\}, P_Número \rightarrow \{P_Nombre, P_Ubicación\}\}$
 - **Paso 2.** Producir relaciones R_1 y R_2
 - $R_1=(E_CUIL, E_Salario, E_Teléfono, D_Número)$
 - $R_2=(P_Número, P_Nombre, P_Ubicación)$
 - **Paso 3.** Generar R_3 adicional con clave de U . Obteniendo finalmente:
 - $R_1=(E_CUIL, E_Salario, E_Teléfono, D_Número)$
 - $R_2=(P_Número, P_Nombre, P_Ubicación)$
 - $R_3=(E_CUIL, P_Número)$

Normalización - Algoritmos Diseño 1 - 3FN (Cont.)

Ejemplo 2.A.

- $U = \{id_Nacional, Provincia, id_Provincial, Zonificación\}$
- $F = \{ FD1: id_Nacional \rightarrow \{Provincia, id_Provincial, Zonificación\}, FD2: \{Provincia, id_Provincial\} \rightarrow \{id_Nacional, Zonificación\}, FD3: Zonificación \rightarrow Provincia \}$
- Abreviaremos $N=id_Nacional, V=Provincia, P=id_Provincial, Z=Zonificación\}$
- $F = \{N \rightarrow VPZ, VP \rightarrow NZ, Z \rightarrow V\}$
- Paso 1.**
 - Aplicando algoritmo de minimal cover, en su paso 2 se obtiene $F = \{N \rightarrow V, N \rightarrow P, N \rightarrow Z, VP \rightarrow N, VP \rightarrow Z, Z \rightarrow V\}$
 - Y en su paso 4, se observa que $N \rightarrow Z$ es redundante (se obtiene por transitividad de $N \rightarrow VP$ y $VP \rightarrow Z$)
 - Así Cubrimiento minimal $G = \{N \rightarrow VP, VP \rightarrow NZ, Z \rightarrow V\}$
- Paso 2.** Producir relaciones R_1, R_2 y R_3
 - $R_1 = (\underline{N}, V, P)$
 - $R_2 = (\underline{V}, P, N, Z)$
 - $R_3 = (\underline{Z}, V)$
- Paso 4.** R_3 y R_1 ambas son proyecciones de R_2 . Por lo tanto, ambas son redundantes
- Así, la descomposición obtenida en 3FN es $R_2 = (\underline{V}, P, N, Z)$
¡Que es idéntica a la original!

Normalización - 2da. Parte

Normalización - Algoritmos Diseño 1 - 3FN (Cont.)

Ejemplo 2.B.

- $U = \{id_Nacional, Provincia, id_Provincial, Zonificación\}$
- $F = \{ FD1: id_Nacional \rightarrow \{Provincia, id_Provincial, Zonificación\}, FD2: \{Provincia, id_Provincial\} \rightarrow \{id_Nacional, Zonificación\}, FD3: Zonificación \rightarrow Provincia \}$
- Abreviaremos $N=id_Nacional, V=Provincia, P=id_Provincial, Z=Zonificación\}$
- $F = \{N \rightarrow VPZ, VP \rightarrow NZ, Z \rightarrow V\}$
- Paso 1.**
 - Aplicando algoritmo de minimal cover, en su paso 2 se obtiene $F = \{N \rightarrow V, N \rightarrow P, N \rightarrow Z, VP \rightarrow N, VP \rightarrow Z, Z \rightarrow V\}$
 - Y en su paso 4, **de manera alternativa**, se observa que $VP \rightarrow Z$ es redundante (se obtiene por transitividad de $VP \rightarrow N$ y $N \rightarrow Z$)
 - También $N \rightarrow V$ es redundante (transitividad de $N \rightarrow Z$ y $Z \rightarrow V$)
 - Así, se obtiene un cubrimiento minimal alternativo:
 $Cubrimiento minimal G = \{N \rightarrow PZ, VP \rightarrow N, Z \rightarrow V\}$
- Paso 2.** Producir relaciones R_1, R_2 y R_3
 - $R_1 = (\underline{N}, P, Z)$
 - $R_2 = (\underline{V}, P, N)$
 - $R_3 = (\underline{Z}, V)$
- Paso 4.** Ninguna es proyecciones de otra. Por lo tanto, es el resultado final **¡Pero difiere del ejemplo anterior!**

Normalización - 2da. Parte

Normalización - Algoritmos Diseño 1 - 3FN (Cont.)

Ejemplo 2.B.

- $U = \{id_Nacional, Provincia, id_Provincial, Zonificación\}$
- $F = \{ FD1: id_Nacional \rightarrow \{Provincia, id_Provincial, Zonificación\}, FD2: \{Provincia, id_Provincial\} \rightarrow \{id_Nacional, Zonificación\}, FD3: Zonificación \rightarrow Provincia \}$
- Abreviaremos $N=id_Nacional, V=Provincia, P=id_Provincial, Z=Zonificación\}$
- $F = \{N \rightarrow VPZ, VP \rightarrow NZ, Z \rightarrow V\}$
- cubrimiento minimal alternativo: $Cubrimiento minimal G = \{N \rightarrow PZ, VP \rightarrow N, Z \rightarrow V\}$
- Resultado.**
 - $R_1 = (\underline{N}, P, Z)$
 - $R_2 = (\underline{V}, P, N)$
 - $R_3 = (\underline{Z}, V)$
- Observaciones.**
 - Se preservan las DFs
 - Se encuentran en BCFN
 - R_2 es redundante en presencia de R_1 y R_3 . Sin embargo, R_2 no se puede eliminar dado que no es proyección de las otras dos relaciones
 - R_2 es importante ya que mantiene las dos CK juntas
 - R_2 mantiene la DF $VP \rightarrow N$ que se perdería si eliminamos dicha relación

Normalización - 2da. Parte

Normalización - Algoritmos Diseño 1 - 3FN

Conclusiones.

- Con el algoritmo, partiendo del mismo conjunto de DFs, se puede generar más de un diseño (Ejemplo 2.A. vs Ejemplo 2.B.)
- En algunos casos, algoritmo puede producir diseños que cumplen con BCFN (incluyendo relaciones que mantienen la preservación de DFs)

Normalización - 2da. Parte

Normalización - Algoritmos Diseño 2 - BCFN

- **Algoritmo Nro. D2** Descomposición en BCFN

Entrada: R universal y un conjunto de DFs F sobre R

1. $D := \{R\}$
2. Mientras $(\exists Q \in D)$ Q no cumple BCFN {
 Seleccionar $Q \in D$ que no cumple BCFN;
 Encontrar DF $X \rightarrow Y$ en Q que no cumple con BCFN;
 Reemplazar Q en D por la siguientes dos relaciones: $(Q - Y)$ y $(X \cup Y)$;
};

- En base a la propiedad NJB (descomposición binaria) y a la Afirmación Nro. 2 D cumple con la propiedad lossless join

Normalización - Algoritmos Diseño 2 - BCFN

- **Ejemplo.**

- $R = \{ \text{Estudiante}, \text{Materia}, \text{Instructor} \}$
- $F = \{ \text{FD1: } \{ \text{Estudiante}, \text{Materia} \} \rightarrow \text{Instructor}, \text{FD2: } \text{Instructor} \rightarrow \text{Materia} \}$

- Aplicando el algoritmo se obtiene

- $R_1 = (\text{Estudiante}, \text{Instructor})$
- $R_2 = (\text{Instructor}, \text{Materia})$

Importante

La teoría de lossless join se basa en la asunción de que no existen valores NULL en los atributos de JOIN

Normalización - Bibliografía

- Capítulo 15 (hasta 15.3 inclusive) Elmasri/Navathe - Fundamentals of Database Systems, 7th Ed., Pearson, 2015.

