

## DISEÑO EN EL MODELO RELACIONAL: NORMALIZACIÓN

{{atributo} = esquema de **relación**} = esquema de **BD Relacional**

- ¿Cómo obtener el esquema de BDR?
  - Transformación del esquema conceptual (diagrama EER)
  - Sentido común del diseñador de BD

¿cómo saber si mi esquema es...

... mejor que el tuyo?

... más apropiado?

... de mayor calidad?

¿Intuición?

Necesitamos **MEDIDAS (FORMALES, si puede ser...)**

Normalización - 1

## DISEÑO EN EL MODELO RELACIONAL

- Existe una Teoría para medir BUENOS esquemas de relación...
  - NIVEL LÓGICO
    - **Interpretación** de **esquema** de relación (base y vistas) y del **significado** de sus **atributos**
    - Buen esquema de relación
      - ⇒ usuario comprende su significado
      - ⇒ consultas correctas
  - NIVEL de ALMACENAMIENTO
    - cómo se **almacenan** y **actualizan** las tuplas de una **relación base**

Normalización - 2

## Pautas de diseño de esquemas de relación

(Pautas informales que sirven como medidas de la calidad)

- Semántica de los atributos de un esquema de relación

**PAUTA 1:** Diseñar esquemas de relación con significado fácil de entender. Evitar combinar atributos de múltiples tipos de entidad e interrelación en una relación. Si una relación proviene de un único tipo de entidad o interrelación suele tener un significado claro; en caso contrario suele ser una mezcla ambigua.

- Reducción de valores redundantes en las tuplas
  - minimizar espacio de almacenamiento ocupado por las relaciones base
  - evitar anomalías de actualización

**PAUTA 2:** Diseñar esquemas de relaciones base sin anomalías de actualización. Si, para incrementar la eficiencia, se permite la posibilidad de anomalías, indicarlas claramente para su tratamiento correcto por parte de las aplicaciones con acceso a los datos (aunque, normalmente, se crearán vistas para las consultas más frecuentes).

Normalización - 3

## Pautas de diseño de esquemas de relación

- Reducción de nulos en las tuplas
  - NULL tiene múltiples significados, lo que dificulta...
    - El entendimiento de la semántica de los atributos y
    - La especificación de operaciones JOIN y agregadas (SUM...)
  - NULL supone el desperdicio de espacio de almacenamiento

**PAUTA 3:** Evitar atributos que puedan contener NULL en las relaciones base. Si no es posible, conseguir que NULL se aplique sólo a casos excepcionales y no a la mayoría de las tuplas.

- Evitar la pérdida de información -- tuplas falsas (espurias) --
  - Esta pauta puede expresada de manera más formal (propiedad de Reunión sin Pérdida --se verá)

**PAUTA 4:** Diseñar las relaciones de forma que puedan ser reunidas (join) mediante condiciones de igualdad sobre atributos Clave Primaria y Clave Externa, para evitar tuplas espurias.

Normalización - 4

## DIFICULTADES EN EL DISEÑO DE BDR

- Propiedades NO DESEABLES que puede tener un diseño de BD incorrecto:
  - Repetición de Información
  - Anomalías de Actualización de la BD
  - Incapacidad para Representar cierta Información
  - Pérdida de Información

Normalización - 5

## DIFICULTADES EN EL DISEÑO DE BDR

Ejemplo 1: información sobre PEDIDOS almacenada en una única relación

PEDIDO(numpedido, codprodu, nomclie, ciudclie, distanc, uds, precio)

- ¿Existe Repetición de Información?
- ¿Hay Anomalías de Actualización?
  - ¿ Qué ocurre si añadimos otro pedido, el nº 27, hecho por el cliente Soriano de Yecla (a 100 km.), de 5 unidades del artículo XM30, cuyo precio por unidad es 25.500 pta. ?
  - ¿ Qué pasa si el cliente Forte de Madrid se traslada a Valencia ?  
¿ Y si el producto XM30 aumenta de precio ?
  - ¿Qué ocurre si eliminamos el artículo XP22 de la relación PEDIDO ?
- ¿Existe Incapacidad para Representar Información?
  - ¿ Cómo representar (almacenar) un cliente que no haya realizado ningún pedido ? ¿ Y un artículo que todavía no haya sido pedido por un cliente ?

Normalización - 6

## DIFICULTADES EN EL DISEÑO DE BDR

### PRESTAMO

<u>nomsuc</u>	<u>ciudsuc</u>	<u>activo</u>	<u>nomcli</u>	<u>numprest</u>	<u>importe</u>
Centro	Arganzuela	1.800.000.000	Santos	P-17	200.000
Moralzarzal	La Granja	420.000.000	Gómez	P-23	400.000
Navacerrada	Aluche	340.000.000	López	P-15	300.000
Centro	Arganzuela	1.800.000.000	Sotoca	P-14	300.000
Becerril	Aluche	80.000.000	Santos	P-93	100.000
Collado	Aluche	1.600.000.000	Abril	P-11	180.000
Navas	Alcalá de H.	60.000.000	Valdivieso	P-29	240.000
Segovia	Cerceda	740.000.000	López	P-16	260.000
Centro	Arganzuela	1.800.000.000	González	P-63	400.000
Navacerrada	Aluche	340.000.000	Rodríguez	P-25	500.000
Galapagar	Arganzuela	1.420.000.000	Amo	P-10	440.000

Normalización - 7

## DIFICULTADES EN EL DISEÑO DE BDR

Ejemplo 2: información sobre PRÉSTAMOS almacenada en una única relación  
PRESTAMO(nomsuc, ciudsuc, activo, nomcli, numprest, importe)

- ¿Existe Repetición de Información?
- ¿Hay Anomalías de Actualización?
  - ¿Qué pasa si añadimos otro préstamo, P-31, por 300.000 pta., para el cliente Fernández, hecho por la sucursal “Navacerrada” (ubicada en “Aluche” y con activo de 340.000.000) ?
  - ¿Qué ocurre si la sucursal de “Madrid” se traslada a “Tres Cantos” ?
  - ¿Qué pasa cuando todos los préstamos de una sucursal se hayan pagado ?
- ¿Existe Incapacidad para Representar Información?
  - ¿Cómo almacenar una sucursal que no haya concedido todavía ningún préstamo?

Normalización - 8

## DIFICULTADES EN EL DISEÑO DE BDR

¿Solución? Descomposición en varios esquemas de relación

- 👁 Una descomposición descuidada puede conducir a otro **diseño incorrecto**: Pérdida de Información

Ejemplo 2: PRESTAMO(nomsuc, ciudsuc, activo, nomcli, numprest, importe)

- Descomposición en DOS esquemas (son proyecciones de PRESTAMO)  
 SUCURSALCLIENTE(nomsuc, ciudsuc, activo, nomcli)  
 PRESTAMOCLIENTE(nomcli, numprest, importe)
- Obtener todas las sucursales con préstamos con importe < 200.000pta.  
 Es necesario RECONSTRUIR la relación PRESTAMO: realizar un NATURAL JOIN entre SUCURSALCLIENTE y PRESTAMOCLIENTE, vía el atributo nomcli

```
SELECT nomsuc FROM SUCURSALCLIENTE SC, PRESTAMOCLIENTE PC
WHERE SC.nomcli=PC.nomcli AND importe < 200000;
```

equivalente a esta otra sentencia:

```
SELECT nomsuc FROM SUCURSALCLIENTE NATURAL JOIN PRESTAMOCLIENTE
WHERE importe < 200000;
```

Normalización - 9

## DIFICULTADES EN EL DISEÑO DE BDR

### SUCURSALCLIENTE

<u>nomsuc</u>	<u>ciudsuc</u>	<u>activo</u>	<u>nomcli</u>
Centro	Arganzuela	1.800.000.000	Santos
Moralzarzal	La Granja	420.000.000	Gómez
Navacerrada	Aluche	340.000.000	López
Centro	Arganzuela	1.800.000.000	Sotoca
Becerril	Aluche	80.000.000	Santos
Collado	Aluche	1.600.000.000	Abril
Navas	Alcalá de H.	60.000.000	Valdivieso
Segovia	Cerceda	740.000.000	López
Centro	Arganzuela	1.800.000.000	González
Navacerrada	Aluche	340.000.000	Rodríguez
Galapagar	Arganzuela	1.420.000.000	Amo

Normalización - 10

## DIFICULTADES EN EL DISEÑO DE BDR

### PRESTAMOCLIENTE

<u>nomcli</u>	<u>numprest</u>	<u>importe</u>
Santos	P-17	200.000
Gómez	P-23	400.000
López	P-15	300.000
Sotoca	P-14	300.000
Santos	P-93	100.000
Abril	P-11	180.000
Valdivieso	P-29	240.000
López	P-16	260.000
González	P-63	400.000
Rodríguez	P-25	500.000
Amo	P-10	440.000

Normalización - 11

## DIFICULTADES EN EL DISEÑO DE BDR

### SUCURSALCLIENTE \* PRESTAMOCLIENTE

<u>nomsuc</u>	<u>ciudsuc</u>	<u>activo</u>	<u>nomcli</u>	<u>numprest</u>	<u>importe</u>
Centro	Arganzuela	1.800.000.000	Santos	P-17	200.000
<b>Centro</b>	<b>Arganzuela</b>	<b>1.800.000.000</b>	<b>Santos</b>	<b>P-93</b>	<b>100.000</b>
Moralzarzal	La Granja	420.000.000	Gómez	P-23	400.000
Navacerrada	Aluche	340.000.000	López	P-15	300.000
<b>Navacerrada</b>	<b>Aluche</b>	<b>340.000.000</b>	<b>López</b>	<b>P-16</b>	<b>260.000</b>
Centro	Arganzuela	1.800.000.000	Sotoca	P-14	300.000
<b>Becerril</b>	<b>Aluche</b>	<b>80.000.000</b>	<b>Santos</b>	<b>P-17</b>	<b>200.000</b>
Becerril	Aluche	80.000.000	Santos	P-93	100.000
Collado	Aluche	1.600.000.000	Abril	P-11	180.000
Navas	Alcalá de H.	60.000.000	Valdivieso	P-29	240.000
<b>Segovia</b>	<b>Cerceda</b>	<b>740.000.000</b>	<b>López</b>	<b>P-15</b>	<b>300.000</b>
Segovia	Cerceda	740.000.000	López	P-16	260.000
Centro	Arganzuela	1.800.000.000	González	P-63	400.000
Navacerrada	Aluche	340.000.000	Rodríguez	P-25	500.000
Galapagar	Arganzuela	1.420.000.000	Amo	P-10	440.000

Normalización - 12

## DIFICULTADES EN EL DISEÑO DE BDR

- Obtener todas las sucursales con préstamos con importes <200.000  
SELECT nomsuc  
FROM SUCURSALCLIENTE NATURAL JOIN PRESTAMOCLIENTE  
WHERE importe < 200000 ;
  - Resultado obtenido (tres tuplas) : ( (Becerril), (Collado), (Centro) )
  - Pero el resultado correcto es : ( (Becerril), (Collado) )
- Hay MÁS tuplas en SUCURSALCLIENTE \* PRESTAMOCLIENTE, pero existe MENOS información: **Descomposición de Reunión CON Pérdida**
  - \*¿Cuál es la causa de esta Pérdida de Información?
  - \*¿Es la siguiente descomposición de Reunión SIN Pérdida?  
SUCURSAL(nomsuc, ciudsuc, activo)  
INFOPRESTAMO(nomsuc, nomcli, numprest, importe)

Normalización - 13

## DEPENDENCIAS FUNCIONALES

Sea R un esquema de relación, cuyos atributos son  $a_1, a_2, \dots, a_n$

En general, una **dependencia funcional (df)**  $\alpha \rightarrow \beta$  es una...

- Restricción entre dos conjuntos de atributos (**descriptores**)  $\alpha$  y  $\beta$ .
- Propiedad inherente al contenido semántico de los datos, que se debe cumplir para cualquier **extensión** de una relación r con el esquema de relación R.
- ✓ Su utilidad es la de describir mejor un esquema de relación, mediante la especificación de restricciones (sobre atributos) que deben cumplirse siempre.
- ✓ Es definida por alguien que conoce (bien) la semántica de los atributos del esquema de relación R.
- ✓ El concepto de dependencia funcional es esencial para el diseño de esquemas de base de datos sin redundancia.

Normalización - 14

## DEPENDENCIAS FUNCIONALES

### DEFINICIÓN FORMAL

Sea  $R$  un esquema de relación, cuyo conjunto de atributos es  $A$ , y sean los descriptores  $\alpha$  y  $\beta$  subconjuntos de  $A$ , se dice que

$$\alpha \rightarrow \beta \text{ donde } \alpha \subseteq A, \beta \subseteq A, \\ \text{si } \forall t_1, t_2 \in r(R) \text{ y } t_1[\alpha] = t_2[\alpha] \\ \text{entonces } t_1[\beta] = t_2[\beta]$$

$\beta$  depende funcionalmente de  $\alpha$  si para cualesquiera dos tuplas  $t_1$  y  $t_2$  (de la extensión de una relación  $r(R)$  cualquiera)

que tengan los mismos valores en los atributos que forman  $\alpha$ , también tienen los mismos valores para los atributos de  $\beta$ .

(También se dice que  $\alpha$  determina funcionalmente a  $\beta$ )

- Nota1: Que  $\alpha \rightarrow \beta$  no significa que *dado  $\alpha$  pueda deducirse el valor de  $\beta$*
- Nota2: Si  $\alpha \rightarrow \beta$ , no sabemos nada acerca de si  $\beta \rightarrow \alpha$  o no

Normalización - 15

## DEPENDENCIAS FUNCIONALES

- Las dfs son propiedades de la semántica de los atributos:
  - Una df no puede ser demostrada, pero sí afirmada por observación de la realidad.
- Son propiedades del esquema de relación (intensión) y no del contenido (extensión)
  - A partir de la observación de una extensión concreta de una relación  $r$  con esquema  $R$ ,  $r(R)$ , sólo se puede deducir que NO existe (no se cumple) cierta dependencia funcional.
- Extensiones permitidas (válidas) y no permitidas
  - Si una dependencia es cierta para un esquema de relación  $R$ , entonces cualquier extensión de una relación  $r(R)$  que no la cumpla, será una extensión NO permitida (no válida).

Normalización - 16

## DEPENDENCIAS FUNCIONALES

- Dado un conjunto de dependencias funcionales DF que R satisface, y sin saber qué tuplas forman su extensión, podemos deducir que R cumple otras dfs (implicadas lógicamente por DF)
- ★ Sea  $R(A, DF)$   
donde  $A=\{a, b, c\}$  y  $DF=\{a \rightarrow b, b \rightarrow c\}$ ,  
entonces la df  $a \rightarrow c$  también se satisface en R
- Normalmente sólo se especifican las dfs obvias para un esquema de relación (las determinadas sin dificultad a partir de la semántica de los atributos de R)
- Esto es esencial para el diseño de buenos esquemas de relación.

Reglas de inferencia para dependencias funcionales

Normalización - 17

## AXIOMAS DE ARMSTRONG

Sean los descriptores  $\alpha, \beta, \gamma$  y  $\delta$ , todos subconjuntos de A

### REGLA DE REFLEXIVIDAD

Si  $\beta \subseteq \alpha$  entonces  $\alpha \rightarrow \beta$

### REGLA DE ADITIVIDAD (AUMENTATIVIDAD)

Si  $\alpha \rightarrow \beta$  entonces  $\alpha\gamma \rightarrow \beta\gamma$

### REGLA DE TRANSITIVIDAD

Si  $\alpha \rightarrow \beta$  y  $\beta \rightarrow \gamma$  entonces  $\alpha \rightarrow \gamma$

- Conjunto de 3 Reglas de inferencia Completo y Correcto

... Pero tedioso de utilizar...

Normalización - 18

## OTRAS REGLAS DE INFERENCIA

... Deducibles a partir de los Axiomas de Armstrong

### REGLA DE PSEUDOTRANSITIVIDAD

Si  $\alpha \rightarrow \beta$  y  $\gamma\beta \rightarrow \delta$  entonces  $\alpha\gamma \rightarrow \delta$

### REGLA DE UNION (aditiva)

Si  $\alpha \rightarrow \beta$  y  $\alpha \rightarrow \gamma$  entonces  $\alpha \rightarrow \beta\gamma$

### REGLA DE DESCOMPOSICIÓN (de proyectividad)

Si  $\alpha \rightarrow \beta\gamma$  entonces  $\alpha \rightarrow \beta$  y  $\alpha \rightarrow \gamma$

Normalización - 19

## CIERRE DE UN CONJUNTO DE dfs

- El cierre de DF (**DF<sup>+</sup>**) es el conjunto de todas las dfs que puedan ser inferidas a partir de DF, aplicando los axiomas de Armstrong.
- Una df  $\alpha \rightarrow \beta$  se infiere de un conjunto de dfs DF de R si toda extensión  $r(R)$  que cumple DF, también cumple  $\alpha \rightarrow \beta$
- Decir que una df  $\alpha \rightarrow \beta \in DF^+$  significa que se deriva o es inferible a partir de las dfs contenidas en DF
- El cálculo de  $DF^+$  suele tener un coste computacional excesivo

Normalización - 20

## CIERRE DE UN CONJUNTO DE dfs

... Por eso su obtención sigue los siguientes pasos:

1. Determinar los conjuntos de atributos  $\alpha$  que son implicantes / determinantes (parte izquierda) de alguna df de DF
  2. Para cada  $\alpha$  aplicar los Axiomas de Armstrong para determinar los atributos que dependen funcionalmente de  $\alpha$
- El paso 2 es el Cálculo del CIERRE DE UN DESCRIPTOR  $\alpha$  sobre un conjunto de dependencias funcionales DF, denotado por  $\alpha^+_{DF}$

Normalización - 21

## CIERRE DE UN CONJUNTO DE ATRIBUTOS

**Cierre de un descriptor  $\alpha$  sobre un conjunto de dependencias funcionales DF**

$$\alpha^+ = \alpha$$

REPETIR HASTA QUE  $\alpha^+$  NO CAMBIE

PARA CADA  $\beta \rightarrow \delta$  EN **DF**

SI  $\beta \subseteq \alpha^+$  Y  $\delta \not\subseteq \alpha^+$

ENTONCES  $\alpha^+ = \alpha^+ \cup \delta$

★ Sea  $R(A, DF)$ , donde  $A=\{a,b,c,g,h,i\}$  y  $DF=\{a \rightarrow b, a \rightarrow c, cg \rightarrow h, cg \rightarrow i, b \rightarrow h\}$

Calcule  $\{ag\}^+_{DF}$

★ Sea  $R(A, DF)$ , donde  $A=\{a,b,c,d,e,f\}$  y  $DF=\{ab \rightarrow c, bc \rightarrow ad, d \rightarrow e, cf \rightarrow b\}$

Calcule  $\{ab\}^+_{DF}$

Normalización - 22

## CIERRE DE UN CONJUNTO DE dfs

**Comprobación de si una dependencia funcional  $\alpha \rightarrow \beta$  se cumple en un conjunto de dependencias funcionales DF**

$$\alpha \rightarrow \beta \in DF^+ \text{ si y sólo si } \beta \subseteq \alpha^+_{DF}$$

★ Sea  $R(A, DF)$ , donde  $A=\{a,b,c,d,e,f\}$  y  $DF=\{ab \rightarrow c, bc \rightarrow ad, d \rightarrow e, cf \rightarrow b\}$   
Compruebe si se cumplen estas dependencias funcionales:

- $ab \rightarrow d$
- $d \rightarrow a$

Normalización - 23

## DETERMINACIÓN DE CLAVES

- Definición formal de Clave (superclave, clave candidata)

Sea  $R(A, DF)$ , el descriptor  $K \subseteq A$  es clave si  $(K)^+_{DF} = A$

Determinación del conjunto K de **superclaves** de  $R(A, DF)$

1.  $K = A$
2. PARA CADA  $\alpha \in K$  (todos los subconjuntos posibles de A)

$$\text{SI } (K - \alpha)^+_{DF} = A, \text{ ENTONCES } K = K - \alpha$$

- Tras esto, puede aplicarse el algoritmo siguiente, para ver cuáles de las superclaves encontradas son claves candidatas (mínimas) en R

Normalización - 24

## DETERMINACIÓN DE SI $\alpha$ ES CLAVE

Determinación de si el descriptor  $\alpha$  es **clave candidata** de  $R(A, DF)$ ?

1. CALCULAR  $\alpha^+_{DF}$
2. SI  $\alpha^+_{DF} = A$  , ENTONCES  $\alpha$  ES SUPERCLAVE,  
SI NO,  $\alpha$  NO ES CLAVE
3. SI  $\alpha$  ES SUPERCLAVE,  
CALCULAR TODOS LOS SUBCONJUNTOS  $\alpha'$  DE  $\alpha$   
REPETIR PARA CADA  $\alpha'$   
SI  $\alpha'^+_{DF} = A$  , ENTONCES  $\alpha$  NO ES CLAVE  
SI NINGUNA  $\alpha'$  CUMPLE LO ANTERIOR,  $\alpha$  ES CLAVE CANDIDATA

- ★ Sea PRESTAMO (nsocio, nomsocio, codlibro, fecha, editorial, pais),  
donde  $DF = \{ \text{nsocio} \rightarrow \text{nomsocio}, \text{nomsocio} \rightarrow \text{nsocio},$   
 $\text{codlibro} \rightarrow \text{editorial}, \text{editorial} \rightarrow \text{pais},$   
 $\{\text{nsocio}, \text{codlibro}\} \rightarrow \text{fecha} \}$   
¿Es clave {nomsocio, codlibro}?

Normalización - 25

## EQUIVALENCIA DE CONJUNTOS DE DEPENDENCIAS FUNCIONALES

- Sean  $DF_1$  y  $DF_2$  conjuntos de dependencias funcionales para  $R$   
se dice que
  - $DF_2$  **se sigue de**  $DF_1$ , o bien que
  - $DF_1$  **recubre a**  $DF_2$si cada extensión de  $R$  que satisface toda  $df$  de  $DF_1$   
también cumple toda  $df$  de  $DF_2$
- O lo que es lo mismo...  
Si cada  $df$  en  $DF_2$  también está en  $DF_1^+$  (puede inferirse de  $DF_1$ )

$DF_1$  y  $DF_2$  son equivalentes si  $DF_1$  recubre a  $DF_2$  y si  $DF_2$  recubre a  $DF_1$

O lo que es lo mismo...

$DF_1$  y  $DF_2$  son equivalentes si  $DF_1^+ = DF_2^+$

Normalización - 26

## EQUIVALENCIA DE CONJUNTOS DE DEPENDENCIAS FUNCIONALES

- Es costoso comprobar que  $DF_1$  y  $DF_2$  son equivalentes, pues supone calcular  $DF_1^+$  y  $DF_2^+$

Así que suele calcularse así:

1. Comprobar si  $DF_1$  recubre a  $DF_2$
2. Comprobar si  $DF_2$  recubre a  $DF_1$

si 1 y 2 tienen como resultado "SI", entonces  $DF_1$  y  $DF_2$  son equivalentes

- Para comprobar si  $DF_1$  recubre a  $DF_2$  es necesario calcular si toda  $df \alpha \rightarrow \beta$  en  $DF_2$  también está en  $DF_1^+$ , es decir
  - a. Calcular  $\alpha^+_{DF_1}$  para cada  $\alpha \rightarrow \beta$  en  $DF_2$
  - b. Ver si  $\beta \subseteq \alpha^+_{DF_1}$si todas las  $df$  de  $DF_2$  cumplen b, entonces  $DF_1$  recubre a  $DF_2$

Normalización - 27

## RECUBRIMIENTO CANÓNICO O MINIMAL

- Conjunto de dependencias funcionales simplificado
- ¿Por qué es interesante utilizar el recubrimiento minimal de un conjunto de dependencias funcionales?

ATRIBUTO EXTRAÑO (RARO) EN UNA DEPENDENCIA FUNCIONAL

- Sea una  $df \alpha \rightarrow \beta$  en  $DF$ ,  
 $a \in \alpha$  es un atributo extraño si la  $df (\alpha - a) \rightarrow \beta$  está en  $DF^+$
- El atributo **a** puede ser eliminado sin modificar el cierre de  $DF$

DEPENDENCIA FUNCIONAL REDUNDANTE DE  $DF$

- La  $df \mathbf{d}$  es una dependencia redundante si puede derivarse de  $\{DF - \mathbf{d}\}$ , aplicando los Axiomas de Armstrong
- La  $df \mathbf{d}$  puede ser eliminada sin modificar el cierre de  $DF$

Normalización - 28

## RECUBRIMIENTO MINIMAL DE DF ( $DF^m$ )

- (0) Transformar toda **df** del conjunto DF en **ELEMENTAL**, es decir:
- Completa
  - Con un único atributo en el implicado (parte derecha)
  - No trivial

**(1) Eliminar ATRIBUTOS EXTRAÑOS**

REPETIR PARA CADA  $\alpha \rightarrow b$  EN DF

$L = \alpha$

REPETIR POR CADA ATRIBUTO a DE  $\alpha$

SI  $b \in (\alpha - a)^+_{DF}$  ENTONCES  $L = L - a$

REEMPLAZAR  $\alpha \rightarrow b$  POR  $L \rightarrow b$

**(2) Eliminar DEPENDENCIAS FUNCIONALES REDUNDANTES**

$H = DF$

REPETIR PARA CADA  $\alpha \rightarrow b$  EN DF

$G = H - \{\alpha \rightarrow b\}$

SI  $b \in \alpha^+_G$  ENTONCES  $H = G$

**(3)  $DF^m = H$**

Normalización - 29

## RECUBRIMIENTO MINIMAL DE DF ( $DF^m$ )

- El orden de los pasos (1) y (2) no puede intercambiarse  
Si se intercambian, no siempre se obtiene el recubrimiento minimal
  - ★ Sea  $R(A, DF)$ , donde  $A = \{a, b, c\}$  y  $DF = \{ab \rightarrow c, c \rightarrow b, a \rightarrow b\}$ .  
Calcule  $DF^m$
- El recubrimiento minimal no es único
  - ★ Sea  $LIBRO(codlibro, isbn, editorial, pais)$ ,  
donde  $DF = \{codlibro \rightarrow isbn editorial,$   
 $isbn \rightarrow codlibro editorial pais,$   
 $editorial \rightarrow pais\}$   
Calcule  $DF^m$
  - ★ Sea  $R(a, b, c)$ ,  
donde  $DF = \{a \rightarrow bc,$   
 $b \rightarrow ac,$   
 $c \rightarrow ab\}$   
Calcule  $DF^m$

Normalización - 30

## NORMALIZACIÓN USANDO DEPENDENCIAS FUNCIONALES

- Uso de un conjunto de dependencias funcionales para diseño correcto de BDR
- Definición de varias FORMAS NORMALES que representen "buenos" diseños

Sea  $R(A, DF)$ , será necesario transformarlo (descomposición) en un conjunto de  $n$  esquemas de relación  $\{ R_i(a_i, DF_i) \} i:1..n$

tal que se cumplan las propiedades siguientes:

- Conservación de la Información
- Conservación de las Dependencias
- Mínima Redundancia de Datos (normalización de relaciones)

Así, las  $R_i$  son **equivalentes** a  $R$ , y **mejores** que  $R$

- ★ Sea  $PRESTAMO(nomsuc, ciudadsuc, activo, nomclie, numprest, importe)$ , donde  $DF = \{ nomsuc \rightarrow activo \text{ ciudadsuc}, numprest \rightarrow importe \}$

La siguiente descomposición cumple las propiedades anteriores:

SUCURSAL(nomsuc, activo, ciudadsuc)  
PRESTAMO(nomsuc, numprest, importe)  
CLIENTE(nomclie, numprest)

Normalización - 31

## CONSERVACIÓN DE LA INFORMACIÓN

La información contenida en  $R$  debe ser la misma que la contenida en las  $R_i$

Para ello, debe conseguirse...

- Conservación de los atributos  $\cup A_i = A$
- Conservación del contenido (tuplas o extensión)  $*r_i = r$ 
  - Para toda  $r$ , extensión de  $R$ , la reunión natural (\*) de las extensiones  $r_i$  de las  $R_i$ , ha de producir la  $r$  de origen (reconstrucción de la extensión original).
  - Si no se conserva el contenido, aparecen **tuplas falsas**
- Una descomposición de reunión con pérdida puede ser detectada para algunas extensiones, pero puede no serlo para otras.
- Es necesario encontrar una forma de descomponer conservando SIEMPRE la información: DESCOMPOSICIÓN en PROYECCIONES INDEPENDIENTES (se verá)

Normalización - 32

## CONSERVACIÓN DE LAS DEPENDENCIAS

La descomposición de  $R$  en los  $R_i$  debe conservar el conjunto de dependencias funcionales  $DF$  de  $R$ ,  
pues cada dependencia funcional es una Restricción de Integridad que refleja la semántica del mundo real

$$(\cup DF_i)^+ = DF^+$$

- El conjunto de dependencias de origen es equivalente al conjunto de dependencias de los esquemas resultantes
- De este modo se consigue que cualquier actualización de la base de datos nunca obtenga una relación ilegal (cuya extensión tras la actualización no cumple todas las dependencias funcionales dadas)

Normalización - 33

## MINIMIZACIÓN DE REDUNDANCIAS

Los  $R_i$  deben estar en una FORMA NORMAL superior a  $R$

- Así se elimina la redundancia, separando los datos en relaciones distintas,
- y por tanto, se evitan las anomalías de actualización y la incapacidad para representar cierta información

FORMAS NORMALES basadas en dependencias funcionales

- 1FN
- 2FN
- 3FN, FNBC

Normalización - 34

## PRIMERA FORMA NORMAL

- Inherente al Modelo Relacional de Datos
- Ningún atributo puede tomar valores compuestos (dominio no atómico), ni multivalorados (más de un valor al mismo tiempo)

### LIBRO (incorrecto: no en 1FN)

<u>codigo</u>	<u>titulo</u>	<u>autor</u>
9090	Sistemas de Bases de Datos	Elmasri, Navathe
8070	Concepción y Diseño de Bases de Datos	de Miguel, Piattini
6050	Fundamentos de Bases de Datos	Ullman

### LIBRO (en 1FN)

<u>codigo</u>	<u>titulo</u>	<u>autor</u>
9090	Sistemas de Bases de Datos	Elmasri
9090	Sistemas de Bases de Datos	Navathe
8070	Concepción y Diseño de Bases de Datos	de Miguel
8070	Concepción y Diseño de Bases de Datos	Piattini
6050	Fundamentos de Bases de Datos	Ullman

Normalización - 35

## SEGUNDA FORMA NORMAL

Nota: atributo principal = atributo que forma parte de una clave

- R está en 2FN si está en 1FN y cada atributo no principal tiene dependencia funcional completa respecto de cada clave.
- No existe un atributo no principal que dependa funcionalmente de parte de una clave

PUBLICA (articulo, revista, numero, pagina, editorial)

DF={ articulo, revista, numero → pagina,  
revista → editorial }

PUBLICA no está en 2FN, ¿por qué?

Normalización - 36

## TERCERA FORMA NORMAL

Varias definiciones:

- R está en 3FN si todo atributo no principal sólo tiene dependencia funcional respecto de las claves.
- Ningún atributo no principal depende funcionalmente de otros atributos no principales (todos los atributos no principales son independientes entre sí)
- No existen dependencias funcionales transitivas respecto de las claves

Definición Formal:

- R(A, DF) está en 3FN respecto a DF si para toda dependencia funcional  $\alpha \rightarrow \beta$  en  $DF^+$  ( $\alpha \subseteq A, \beta \subseteq A$ ), se cumple al menos una de las siguientes condiciones:
  - ✓  $\alpha \rightarrow \beta$  es una dependencia funcional trivial, es decir,  $\beta \subseteq \alpha$
  - ✓  $\alpha$  es una superclave del esquema R, es decir  $\alpha^+ = A$
  - ✓ Cada atributo a en  $\beta - \alpha$  es principal (está contenido en alguna clave candidata de R)

SOCIO (dni, ciudad, pais)

DF = { dni  $\rightarrow$  ciudad, ciudad  $\not\rightarrow$  dni,  
ciudad  $\rightarrow$  pais, dni  $\rightarrow$  pais }

SOCIO no está en 3FN, ¿por qué?

Normalización - 37

## FORMA NORMAL DE BOYCE y CODD

- Definición más estricta de la 3FN

Nota: determinante = implicante o parte izquierda de una df

- R(A, DF) está en FNBC respecto a DF si para toda dependencia funcional  $\alpha \rightarrow \beta$  en  $DF^+$  ( $\alpha \subseteq A, \beta \subseteq A$ ), se cumple al menos una de estas condiciones:
  - ✓  $\alpha \rightarrow \beta$  es una dependencia funcional trivial, es decir,  $\beta \subseteq \alpha$
  - ✓  $\alpha$  es una superclave del esquema R, es decir  $\alpha^+ = A$
- Es decir, R está en FNBC si todo determinante contiene una clave candidata

Observaciones:

- La FNBC no requiere que todas las claves aparezcan como determinantes.
- Si R está en 3FN y sólo tiene una clave, o bien  
R tiene varias claves pero ninguna de ellas es compuesta, o bien  
R tiene varias claves compuestas pero no solapadas,  
entonces R está en FNBC
- Es decir, si R tiene claves solapadas puede estar en FNBC o puede no estarlo  
(véase siguiente ejemplo)

Normalización - 38

## FORMA NORMAL DE BOYCE y CODD

### ARTICULO (codart, titulo, revista, numero, pagina)

Un título nunca se repite en una misma revista.  
Un artículo puede publicarse en varias revistas

**DF** = { codart → titulo;  
titulo → codart;  
codart, revista, numero → pagina;  
titulo, revista, numero → pagina;  
pagina, revista, numero → codart, titulo }

Está en 3FN, pero NO en FNBC

CK1 = { revista, numero, codart }  
CK2 = { revista, numero, titulo }  
CK3 = { revista, numero, pagina }

### ARTICULO2 (codart, revista, numero, pagina)

**DF** = { codart, revista, numero → pagina;  
pagina, revista, numero → codart }

Está en 3FN, y también en FNBC

CK1 = { revista, numero, codart }  
CK2 = { revista, numero, pagina }

Normalización - 39

## Algunas consideraciones

Intente comprobar la veracidad de estas afirmaciones:

- Todo esquema de relación R con una clave simple (un solo atributo), está en 2FN
  - Si en R todos los atributos son principales, R está en 3FN
  - Todo esquema R con sólo dos atributos está siempre en FNBC
  - Todo esquema con 3 atributos y una única clave compuesta por dos atributos, está en FNBC
- Lo más correcto es tener los esquemas en FNBC, pero si no es posible, basta con llegar a la 3FN.
- 1FN y 2FN NO son buenos diseños, sino "escalones" para llegar a 3FN o FNBC
- El proceso de normalización disminuye la redundancia (y por tanto el desperdicio de espacio y las anomalías de actualización), pero penaliza las consultas (pues necesitan joins cuyo coste es elevado).

Normalización - 40

## ENFOQUES DE DISEÑO RELACIONAL: ANÁLISIS y SÍNTESIS

- ANÁLISIS o DESCOMPOSICIÓN (diseño descendente)
  - Parte del esquema de RELACIÓN UNIVERSAL, R y de su conjunto de DEPENDENCIAS, DF
  - Descompone R sucesivamente en proyecciones que
    - conservan la información
    - conservan las dependencias
  - Los esquemas resultantes cada vez ...
    - Tienen menos atributos (menor grado)
    - Están en FN más avanzadas
  - Fin del proceso ...
    - Al llegar a la FN deseada
    - Si seguir implica la pérdida de dependencias
- Separa información referente a conceptos distintos
- Árboles de descomposición

Normalización - 41

## ENFOQUES DE DISEÑO RELACIONAL: ANÁLISIS y SÍNTESIS

- SÍNTESIS RELACIONAL (diseño ascendente)
  - Obtiene relaciones a partir de
    - atributos
    - dependencias funcionales entre dichos atributos
- Agrupar información referente a un mismo concepto

### COMPARACIÓN ANÁLISIS - SÍNTESIS

- Tienen el mismo objetivo
- Basados en los conceptos de **dependencias** y **recubrimiento minimal**
- La **síntesis sólo** considera **dependencias funcionales**
- El **análisis** considera, además de las **dependencias funcionales**, las dependencias **multivaloradas** y dependencias de **reunión** o **combinación**

Normalización - 42

## ANÁLISIS (DISEÑO POR DESCOMPOSICIÓN)

Examinar una estructura relacional R para...

- 1. Determinar su **nivel de normalización**
- 2. Descomponer R en nuevas estructuras relacionales (**proyecciones**)  
sin redundancias ni anomalías  
si R no está en 3FN ni en FNBC

### 1. DETERMINAR EL NIVEL DE NORMALIZACIÓN DE R

\* CÓMO DETERMINAR SI UN ESQUEMA DE RELACIÓN ESTÁ EN **3FN**

- .a. Encontrar todas las claves para ...
- .b. Determinar componentes de dichas claves (atributos principales) y
- .c. Comprobar que ningún atributo no principal, depende transitivamente de una clave

\* CÓMO DETERMINAR SI UN ESQUEMA DE RELACIÓN ESTÁ EN **FNBC**

- .a. Comprobar que todo determinante (parte izquierda de una df) es clave  
-- menor coste computacional

Normalización - 43

## ANÁLISIS

### 2. DESCOMPONER EL ESQUEMA R (A, DF)

Si R no está en 3FN ni en FNBC, hemos de sustituirlo por un conjunto de **proyecciones**  $R_i (A_i, DF_i)$

.a. Sin pérdida de información:

$$R = R_1 * \dots * R_n, \text{ para toda extensión de R}$$

.b. Sin pérdida de dependencias funcionales

$$(\cup DF_i)^+ = DF^+$$

Es decir, en proyecciones **INDEPENDIENTES**  
(descomposición sin pérdidas)

Normalización - 44

## ANÁLISIS

### DESCOMPOSICIÓN DE R EN PROYECCIONES INDEPENDIENTES

Sea R una relación y  $R_1, R_2$  dos de sus proyecciones

Entonces  $R_1$  y  $R_2$  son independientes si y sólo si

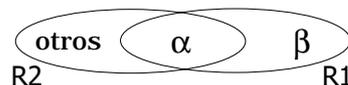
- .a. Sus atributos comunes son la CLAVE PRIMARIA de, al menos, una de las dos proyecciones
- .b. Cada DEPENDENCIA FUNCIONAL en R puede deducirse de las dependencias funcionales de  $R_1$  y  $R_2$

Guía en el proceso de Descomposición en Proyecciones Independientes:  
Utilizar dependencias funcionales que violen la FN deseada (FNBC)

Normalización - 45

## ANÁLISIS

- Tomar  $\alpha \rightarrow \beta$  de  $DF^+$  no trivial, donde  $\alpha$  no es superclave
- Obtener dos proyecciones de  $R(A, DF)$   
 $R_1 (\alpha, \beta, DF_1)$  y  
 $R_2 (A - \beta, DF_2)$ ,  
donde  $DF_1$  y  $DF_2$  son conjuntos de dfs sobre  $R_1$  y  $R_2$
- Si  $R_1$  o  $R_2$  no están en FNBC, descomponerlas de nuevo



La descomposición resultante es de reunión sin pérdida,  
pero no siempre preserva las dependencias

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

Normalización - 46

## ANÁLISIS

### DETERMINACIÓN DE LAS DEPENDENCIAS FUNCIONALES QUE CUMPLE CADA PROYECCIÓN RESULTANTE DE UNA DESCOMPOSICIÓN

Sea  $R(A, DF)$  y sea  $R1(A1, DF1)$  una de las relaciones en las que  $R$  se ha descompuesto

- Considerar cada descriptor  $\alpha$ , subconjunto de  $A1$
- Calcular  $\alpha^+$ , respecto del conjunto de dependencias  $DF$  (#)
- Para cada atributo  $a$  tal que
  - $a$  es un atributo de  $R1$  ( $a \in A1$ )
  - $a$  está contenido en  $\alpha^+$  ( $a \in \alpha^+$ )
  - $a$  no está contenido en  $\alpha$  ( $a \notin \alpha$ )la dependencia funcional  $\alpha \rightarrow a$  se cumple en  $R1$ , es decir  $(\alpha \rightarrow a) \in DF1$

(#) No es necesario determinar el cierre de ...

- todo el conjunto de atributos de  $R1$  (es decir, si  $\alpha = A1$ )
- los  $\alpha$  que no contengan la parte izquierda de alguna dependencia
- los  $\alpha$  que contengan un atributo que no sea parte izquierda de alguna dependencia

Normalización - 47

## ANÁLISIS

- Las DOS proyecciones independientes resultantes cumplen uno de estos PRINCIPIOS de DESCOMPOSICIÓN SIN PÉRDIDAS:
  - $R_1 \cap R_2 \rightarrow R_1 - R_2$  o bien
  - $R_1 \cap R_2 \rightarrow R_2 - R_1$
- Para las TRES PRIMERAS FORMAS NORMALES siempre es posible descomponer las relaciones en DOS proyecciones independientes
- Esto no siempre es posible para llegar a FNBC, pues a veces se pierde información y/o dependencias funcionales

### Ejemplos

1. PROYECTA(pelicula, cine, ciudad)  
pelicula, ciudad  $\rightarrow$  cine  
cine  $\rightarrow$  ciudad
2. NOTA(dniProf, nomProf, dniAlu, califc)  
dniProf  $\rightarrow$  nomProf  
nomProf  $\rightarrow$  dniProf  
dniProf, dniAlu  $\rightarrow$  califc

Normalización - 48

## ANÁLISIS

### PROCESO DE DESCOMPOSICIÓN de $R(A, DF)$

1. Determinar el recubrimiento minimal  $DF^m$
2. Determinar las claves de R
3. Obtener Proyecciones Independientes hasta que se ha llegado a la FN deseada o si seguir descomponiendo supusiera pérdidas

### DESCOMPOSICIÓN en la FNBC de $R(A, DF)$

1. Determinar el recubrimiento minimal  $DF^m$
2. Determinar las claves de R
3. Obtener Proyecciones Independientes hasta que todo determinante es una clave (se ha llegado a la FNBC) o seguir implique pérdidas

#### Ejemplo

$R(A, DF)$ , donde  $A = \{ a, b, c, d, e, f, g \}$  y  $DF^m = \{ b \rightarrow ac, e \rightarrow d, d \rightarrow f \}$

Normalización - 49

## ANÁLISIS: ejemplo 1

Aplicar el algoritmo al siguiente esquema de relación para llevarlo a la FNBC:

$PRESTAMO(nomsuc, ciudsuc, activo, nomclie, numprest, importe)$   
donde  $DF = \{ nomsuc \rightarrow activo, ciudsuc; numprest \rightarrow importe, nomsuc \}$   
y  $CK = \{ nomclie, numprest \}$

- paso 1.  
 $R1(nomsuc, ciudsuc, activo);$   
 $DF1\{nomsuc \rightarrow activo, ciudsuc\}; CK1=\{nomsuc\}$   
 $R2(nomsuc, nomclie, numprest, importe)$   
 $DF2=\{numprest \rightarrow importe, nomsuc\}; CK2=\{nomclie, numprest\}$
- paso 2.  
 $R1$  está en FNBC  
 $R2$  no. Así que se substituye por  
 $R3(nomsuc, numprest, importe);$   
 $DF3=\{numprest \rightarrow importe, nomsuc\}; CK3=\{numprest\}$   
 $R4(nomclie, numprest); DF4= \emptyset; CK4=\{nomclie, numprest\}$
- paso 3.  
 $R3$  y  $R4$  están en FNBC

El esquema de BD es  $\{ R1, R3, R4 \}$  y está en FNBC

Normalización - 50

## ANÁLISIS: ejemplo 2

Aplicar el algoritmo al siguiente esquema de relación para llevarlo a la FNBC :

-Banqueros personales de cada cliente en cada sucursal bancaria-

BANQUERO(nomsuc, nomclie, nombanquero)

donde DF={ nomsuc → nomclie;

nomsuc, nomclie → nombanquero}

y CK = {nomsuc, nomclie}

• paso 1.

R1(nombanquero, nomsuc);

DF1{nombanquero → nomsuc}; CK1={nombanquero}

R2 (nomclie, nombanquero)

DF2= ∅; CK2={nomclie, nombanquero}

• paso 2.

R1 y R2 están en FNBC

Esquema de BD = { R1, R2 }, en FNBC, pero sólo preserva nombanquero→nomsuc (además de las df triviales) y no conserva nomsuc,nomclie→nombanquero

La violación de esta df no se detectará, a menos que se calcule la reunión

Normalización - 51

## SÍNTESIS

**PROCESO DE SÍNTESIS RELACIONAL** para R(A, DF),

1. Determinar el recubrimiento minimal  $DF^m$

2. Agrupar las Dependencias Funcionales de  $DF^m$  con el mismo determinante

Es decir, agrupar las  $\alpha \rightarrow A_1, \alpha \rightarrow A_2 \dots \alpha \rightarrow A_n$ , para cada antecedente  $\alpha$

3. Formar una  $R_i$  para cada grupo, con los atributos que aparecen en el grupo y las dependencias funcionales de  $DF^m$  con  $\alpha$  como antecedente

$R_i(\alpha \cup A_1 \cup \dots \cup A_n)$  con  $DF_i = \{ \alpha \rightarrow A_1, \dots, \alpha \rightarrow A_n \}$

4. Si existen atributos que no aparecen en ningún determinante ni implicado, formar una  $R_i$  con ellos, sin dependencias funcionales asociadas.

5. Si ninguna  $R_i$  contiene una clave de la R original, añadir una relación con los atributos que forman tal clave de R

Obtiene las  $R_i$  en 3FN

Siempre es posible encontrar una descomposición de reunión sin pérdida que preserve las dependencias y esté en 3FN

Normalización - 52

## SÍNTESIS: ejemplo

Aplicar el algoritmo al siguiente esquema de relación para llevarlo a la 3FN  
-Banqueros personales de cada cliente en cada sucursal bancaria-

BANQUERO(nomsuc, nomclie, nombanquero, numoficina)

donde DF={nombanquero → nomsuc, numoficina;

nomsuc, nomclie → nombanquero}

y CK = {nomsuc, nomclie}

- paso 1.  
DF<sup>m</sup> coincide con DF
- pasos 2 y 3.  
R1(nombanquero, nomsuc, numoficina);  
DF1{nombanquero → nomsuc, numoficina}; CK1={nombanquero}  
R2 (nomclie, nomsuc, nombanquero)  
DF2= {nomsuc, nomclie → nombanquero}; CK2={nomsuc, nomclie}
- paso 4. Todos los atributos han sido colocados
- paso 5. R2 ya contiene una clave candidata de BANQUERO, luego el proceso acaba  
El esquema de BD es { R1, R2 } y está en 3FN

Normalización - 53

## DEPENDENCIAS MULTIVALORADAS - dmv

- Es posible que todavía exista redundancia en esquemas de relaciones en FNBC
- Una dependencia multivalorada proviene de dos o más atributos multivalorados que son independientes entre sí
  - a. ESTUDIANTERESIDENTE(dni, estudios, actividad)
  - b. ACTOR(nombreactor, calle, ciudad, titulopeli, añopeli)
- La única forma de expresar la independencia de los conjuntos de valores ..
  - a1) direcciones de los actores y a2) películas en las que han participado
  - b1) estudios y b2) actividades de los estudiantes.. de los dos conjuntos de atributos ..
  - a. {calle, ciudad} y {titulopeli, añopeli}
  - b. {estudios} y {actividad}.. es hacer que
  - a. para cada actor, cada dirección aparezca con cada una de las películas, y
  - b. para cada estudiante, cada valor de estudios aparezca con cada valor de actividad
- Las dmv son una consecuencia de la transformación de una R en la 1FN cuando existen varios atributos multivalorados e independientes entre sí

Normalización - 54

## DEPENDENCIAS MULTIVALORADAS

Sea el esquema de relación  $R(A, D)$  y dos descriptores (conjuntos de atributos)  $\alpha$  y  $\beta$ ,

$\alpha \twoheadrightarrow \beta$  ( $\alpha$  **multidetermina**  $\beta$ ) si y sólo si,

para cada par de tuplas  $t_1$  y  $t_2$  de  $R$  tales que

$$t_1[\alpha] = t_2[\alpha]$$

es posible encontrar un par de tuplas  $t_3$  y  $t_4$  en  $R$  que cumplen

$$1. t_1[\alpha] = t_2[\alpha] = t_3[\alpha] = t_4[\alpha]$$

$$2. t_3[\beta] = t_1[\beta] \quad \text{y} \quad t_4[\beta] = t_2[\beta]$$

$$3. t_3[A - \alpha - \beta] = t_2[A - \alpha - \beta] \quad \text{y} \quad t_4[A - \alpha - \beta] = t_1[A - \alpha - \beta]$$

\*Para cada valor de  $\alpha$ , existe un conjunto de 0 o más valores de  $\beta$ , independientemente de los valores del resto de atributos ( $A - \alpha - \beta$ )

\*Que se verifique  $\alpha \twoheadrightarrow \beta$  sólo depende de  $\alpha$  y de  $\beta$ , pero que se cumpla  $\alpha \twoheadrightarrow \beta$  depende, además, del resto de los atributos (las dmv dependen del contexto)

\*  $\alpha \twoheadrightarrow \beta$  significa que siempre que aparezcan dos tuplas con distintos valores de  $\beta$ , pero el mismo valor en  $\alpha$ , los valores de  $\beta$  deberán repetirse con cada valor distinto de  $A - \alpha - \beta$  que ocurra con dicho valor de  $\alpha$

nombreactor  $\twoheadrightarrow$  calle, ciudad

Normalización - 55

## DEPENDENCIAS MULTIVALORADAS

### REGLAS DE INFERENCIA PARA dmV

#### Regla de DEPENDENCIAS TRIVIALES

Si  $R$  cumple  $\alpha \twoheadrightarrow \beta$  entonces cumple

·  $\alpha \twoheadrightarrow \gamma$ , donde  $\gamma \subseteq \beta$  pues se han quitado componentes de  $\beta$  que también estaban en  $\alpha$

·  $\alpha \twoheadrightarrow \delta$ , donde  $\beta \subseteq \delta$  pues se han añadido componentes a  $\beta$  que también estaban en  $\alpha$

siempre que  $\delta - \beta \subseteq \alpha$

#### REGLA DE AUMENTO

si  $R$  cumple  $\alpha \twoheadrightarrow \beta$  y  $\gamma \subseteq \delta$ , entonces cumple  $\alpha\delta \twoheadrightarrow \beta\gamma$

#### REGLA TRANSITIVA PARA DMV

si  $R$  cumple  $\alpha \twoheadrightarrow \beta$  y  $\beta \twoheadrightarrow \gamma$ , entonces cumple  $\alpha \twoheadrightarrow \gamma$

#### REGLA DE REPLICACIÓN

Si  $R$  cumple  $\alpha \twoheadrightarrow \beta$ , entonces cumple  $\alpha \twoheadrightarrow \beta$

Normalización - 56

## DEPENDENCIAS MULTIVALORADAS

### REGLA DE COMPLEMENTO

Si  $R$  cumple  $\alpha \twoheadrightarrow \beta$ , también cumple  $\alpha \twoheadrightarrow A-\alpha-\beta$

### REGLA DE COMBINACIÓN PARA dmv Y df

Si  $R$  cumple que  $\alpha \twoheadrightarrow \beta$  y  $\exists \delta / \delta \cap \beta = \emptyset, \delta \rightarrow \gamma$  y  $\gamma \subseteq \beta$ , entonces  $\alpha \rightarrow \gamma$

### DEPENDENCIA MULTIVALORADA TRIVIAL

Sean  $\alpha = \{a_1, a_2 \dots a_n\}$ ,  $\beta = \{b_1, b_2 \dots b_m\}$  descriptores de  $R(A, D)$ ,

$\alpha \twoheadrightarrow \beta$  es una dmv es trivial si y sólo si cumple

- los  $b_i$  están entre los  $a_i$  (es decir  $\beta \subseteq \alpha$ ), o bien
- todos los atributos de  $R$  están entre los  $a_i$  y los  $b_i$  (es decir  $\alpha \cup \beta = A$ )

dmv no trivial: nombreactor  $\twoheadrightarrow$  calle, ciudad

Normalización - 57

## LA CUARTA FORMA NORMAL (4FN)

- Es una generalización de la FNBC
- Elimina las dmv no triviales

$R(A, D)$  está en **4FN** si para toda dmv no trivial  $\alpha \twoheadrightarrow \beta$ , se cumple que  $\alpha$  es una superclave de  $R$

### DESCOMPOSICIÓN EN LA 4FN

- Si el esquema  $R$  no está en 4FN, por contener una dmv  $\alpha \twoheadrightarrow \beta$ , donde  $\alpha$  no contiene una clave de  $R$ , se debe dividir  $R$  en dos esquemas:
  - $R_1$ , con todos los atributos de  $\alpha$  y los de  $\beta$
  - $R_2$ , con todos los atributos de  $\alpha$  y el resto de atributos  $A-\alpha-\beta$
- ✓ Asignar adecuadamente a cada esquema las dependencias funcionales y multivaloradas
- Esto cumple los principios de descomposición sin pérdidas pues  $\alpha \twoheadrightarrow \beta$  y  $\alpha \twoheadrightarrow A-\alpha-\beta$

$R_1$ (nombreactor, calle, ciudad)

$R_2$ (nombreactor, titulopeli, añoopeli)

Normalización - 58

## DEPENDENCIAS DE REUNIÓN o COMBINACIÓN

- Hasta ahora, las df y las dmv permitían descomposición sin pérdidas de una relación R en dos de sus proyecciones
- Existen relaciones para las que **no existe** una **descomposición sin pérdidas en dos esquemas** de relación, **pero sí que dé tres o más esquemas**
- Estos casos se manejan con las **dependencias de reunión o combinación** (join dependences) y la **5FN**
- Son casos raros y difíciles de detectar en la práctica
- Una dependencia de combinación se establece entre una relación R y sus proyecciones  $R_1, R_2, \dots, R_n$  y se denota  **$*$ ( $R_1, R_2, \dots, R_n$ )**
- Restringe el conjunto de relaciones legales a aquellas para las que  $R_1, R_2, \dots, R_n$  es una descomposición de reunión sin pérdida

R tiene la propiedad de **dependencia de reunión o combinación** con respecto de sus proyecciones  $R_1, R_2, \dots, R_n$  si y sólo si se cumple que

$$R = R_1 * R_2 * \dots * R_n$$

Una **DEPENDENCIA DE REUNIÓN** es **TRIVIAL** si alguno de los  $R_i$  es la misma R

Normalización - 59

## LA QUINTA FORMA NORMAL (5FN)

- Elimina las dependencias de reunión no triviales

$R(A,D)$  está en **5FN** respecto a un conjunto D de dependencias funcionales, multivaluadas y de reunión si y sólo si para toda dependencia de reunión no trivial implicada por D, toda  $R_i$  es superclave de R

$R(A,D)$  está en **5FN** si y sólo si toda dependencia funcional, multivaluada y de reunión es consecuencia de las claves candidatas

### DESCOMPOSICIÓN EN LA 5FN

- Si R no está en 5FN, por contener una dr sobre sus descriptores  $\alpha_1, \alpha_2, \dots, \alpha_p$ , se puede descomponer sin pérdida de información en  $p$  proyecciones independientes, de forma que

$$R = R_1(\alpha_1) * R_2(\alpha_2) * \dots * R_p(\alpha_p)$$

- Cada  $R_i$  estará en 5FN

Ejemplo

SUMINISTRAR (nomProveedor, nomComponente, nomProyecto)

Normalización - 60

## FORMA NORMAL de DOMINIO/CLAVE (FNDK)

[Fagin1981]

- Concebida con la idea de definir la Forma Normal definitiva, que considera todos los tipos de dependencias y restricciones

$R(A,D)$  está en **FNDK** si y sólo si todas las restricciones y dependencias se pueden imponer con sólo hacer cumplir las restricciones de dominio y las de clave especificadas sobre cada relación

$R(A,D)$  está en **FNDK** si y sólo si toda restricción de  $R$  es una consecuencia lógica de las restricciones de dominio y de clave de  $R$

- Si se lleva un esquema  $R$  a la FNDK, seguro que nunca tendrá anomalías, ... pero... ¿cómo llevar  $R$  a la FNDK?
- Además, no se ha demostrado si puede alcanzarse la FNDK para cualquier esquema de relación, ni bajo qué condiciones se puede conseguir

Normalización - 61

## PROCESO COMPLETO DE DESCOMPOSICIÓN

... considerando dependencias funcionales, multivaloradas y de reunión...

Sea el esquema relacional  $R(A, DEP)$

1. Determinar el recubrimiento minimal  $DEP^m$
2. Determinar las claves de  $R$
3. Eliminar las dependencias multivaloradas mediante descomposición en proyecciones independientes
4. Considerar las dependencias funcionales para obtener proyecciones independientes, hasta que no se pueda continuar, es decir...
  - si todo determinante de las dependencias funcionales es una clave (FNBC), o
  - si seguir implica pérdidas de información o de dependencias funcionales
5. Para toda relación resultante, comprobar que no existen dependencias de reunión no implicadas por una clave.  
Si existen, eliminarlas con la descomposición en proyec. independientes

Normalización - 62