



## Unidad Didáctica 4:

### Diseño de Bases de Datos Relacionales

## Parte 3:

### Teoría de la Normalización

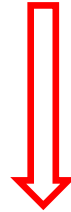
## Documento base:

### UD4.3.pdf

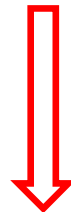
# 5 Teoría de la Normalización

---

Antes de considerar definitivo el esquema relacional obtenido con las transformaciones presentadas, este esquema debe ser revisado para comprobar que se encuentra adecuadamente diseñado.



Normalizando el esquema relaciona aplicando la Teoría de la Normalización.



*Proceso durante el cual los esquemas de relación insatisfactorios se descomponen repartiendo sus atributos entre esquemas de relación más pequeños que poseen propiedades deseables.*

# 5.1 Conceptos Previos

---

- Dependencia Funcional (completa o no).
- Diagrama de Dependencias Funcionales.
- Clave de una relación.
- Atributo Primo.

# 5.1 Conceptos Previos

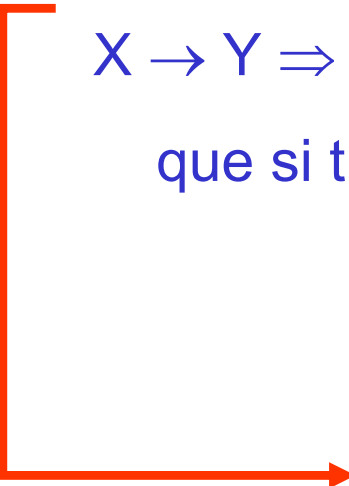
---

## ➤ Dependencia Funcional (completa o no)

□  $A = \{A_1, \dots, A_n\}$  conjunto de atributos del esquema  $R$

Una dependencia funcional entre  $X$  e  $Y$  ( $X \subseteq A, Y \subseteq A$ )

$X \rightarrow Y \Rightarrow$  para cualquier par de tuplas posibles de  $R$ ,  $t_1$  y  $t_2$ , se cumple que si  $t_1[X]$  es igual a  $t_2[X]$ , entonces  $t_1[Y]$  es igual a  $t_2[Y]$ .



Para un valor de  $X$ ,  $Y$  sólo puede tomar un valor posible

# 5.1 Conceptos Previos

---

- **Reglas de inferencia:** permiten deducir todas las dependencias funcionales a partir de un conjunto dado.

Sean  $W, X, Y, Z$  subconjuntos de atributos de una relación:

- Axiomas de Amstrong:
  - ✓ R1: **Reflexiva:** si  $Y \subseteq X \Rightarrow X \rightarrow Y$
  - ✓ R2: **Aumento:** si  $X \rightarrow Y \Rightarrow X \cup Z \rightarrow Y$  y  $X \cup Z \rightarrow Y \cup Z$
  - ✓ R3: **Transitiva:** si  $X \rightarrow Y$  e  $Y \rightarrow Z \Rightarrow X \rightarrow Z$
- Reglas derivadas:
  - ✓ R4: **Unión:** si  $X \rightarrow Y$  y  $X \rightarrow Z \Rightarrow X \rightarrow Y \cup Z$
  - ✓ R5: **Descomposición:** si  $X \rightarrow Y \cup Z \Rightarrow X \rightarrow Y$  y  $X \rightarrow Z$
  - ✓ R6: **Pseudotransitiva:** si  $X \rightarrow Y$  e  $Y \cup W \rightarrow Z \Rightarrow X \cup W \rightarrow Z$

# 5.1 Conceptos Previos

---

## ➤ Dependencia Funcional **completa**

Una dependencia funcional entre dos conjuntos de atributos  $X \rightarrow Y$  es completa si la eliminación de cualquier atributo  $A_i$  de  $X$  hace que la dependencia deje de existir, es decir si  $\forall A_i / A_i \in X$  se cumple que  $Y$  no depende funcionalmente de  $(X - \{A_i\})$ .

# 5.1 Conceptos Previos

---

## ➤ Diagrama de Dependencias Funcionales

Representación gráfica de las dependencias. Utilizan cajas para enmarcar los atributos o conjuntos de atributos y flechas para denotar la dependencia funcional. Normalmente sólo se representan las dependencias funcionales completas.

# 5.1 Conceptos Previos

---

## ➤ Clave de una relación

Sea  $R$  un esquema de relación cuyo conjunto de atributos es  $A = \{A_1, A_2, \dots, A_n\}$ , y sea  $C$  un subconjunto de atributos de ese esquema ( $C \subseteq A$ ); se dice que  $C$  es una clave de  $R$  si  $C$  es la clave primaria de  $R$  o bien si  $C$  tiene una restricción de unicidad.



Todos los atributos de una relación dependen funcionalmente de cada clave que hay en la relación.



# 5.1 Conceptos Previos

## ➤ Dependencias funcionales y claves:

A partir de una clave se pueden deducir dependencias funcionales:

Sea  $R(A_1, \dots, A_n)$  un esquema de relación:

1. Si  $\{A_1, A_2\}$  es una clave de  $R$ , entonces el resto de atributos de  $R$  dependen funcionalmente de  $\{A_1, A_2\}$ :

✓  $\{A_1, A_2\} \rightarrow \{A_3, \dots, A_n\}$ : al ser  $\{A_1, A_2\}$  clave, cada valor del par sólo puede aparecer una vez en la relación y por tanto  $\{A_3, \dots, A_n\}$  sólo puede tomar un valor.

✓  $\{A_1, A_2\} \rightarrow \{A_3, \dots, A_n\} + R5 \Rightarrow \{A_1, A_2\} \rightarrow \{A_3\}$

...

$\{A_1, A_2\} \rightarrow \{A_n\}$

# 5.1 Conceptos Previos

---

## ➤ Dependencias funcionales y claves:

A partir de dependencias funcionales se pueden deducir claves:

Sea  $R(A_1, \dots, A_n)$  un esquema de relación:

2. Si  $\{A_1, A_2\} \rightarrow \{A_3, \dots\}$  y  $\{A_1, A_2\} \rightarrow \{A_n\}$  entonces se puede afirmar que  $\{A_1, A_2\}$  es una clave de  $R$

✓  $\{A_1, A_2\} \rightarrow \{A_3, \dots\}$  y  $\{A_1, A_2\} \rightarrow \{A_n\} + R4$

$\Rightarrow \{A_1, A_2\} \rightarrow \{A_3, \dots, A_n\}$

✓  $\{A_1, A_2\} \rightarrow \{A_3, \dots, A_n\}$ :  $\{A_3, \dots, A_n\}$  sólo puede tomar un valor para un valor de  $\{A_1, A_2\}$  por lo tanto  $\{A_1, A_2\}$  no se puede repetir en dos filas de  $R$  por lo que es una clave.

# 5.1 Conceptos Previos

---

## ➤ Atributo Primo

Sea  $R$  un esquema de relación cuyo conjunto de atributos es  $A = \{A_1, A_2, \dots, A_n\}$ , un atributo  $A$  se dice que es primo si forma parte de alguna clave de  $R$ .

# Ejemplo

---

`Ha_escrito(DNI, nombre, ISBN, título, euros)`

“el escritor de D.N.I. *DNI* que se llama *nombre* ha escrito, solo o en colaboración, el libro de código *ISBN* y de título *título* y ha recibido por ese trabajo la cantidad de *euros euros*”.

# Ejemplo

---

`Ha_escrito(DNI, nombre, ISBN, título, euros)`

- |  |   |
|--|---|
| 1. $\{DNI\} \rightarrow \{nombre\}$                      | 11. $\{ISBN, nombre\} \rightarrow \{título\}$             |
| 2. $\{DNI, ISBN\} \rightarrow \{nombre\}$                | 12. $\{ISBN, euros\} \rightarrow \{título\}$              |
| 3. $\{DNI, título\} \rightarrow \{nombre\}$              | 13. $\{ISBN, DNI, nombre\} \rightarrow \{título\}$        |
| 4. $\{DNI, euros\} \rightarrow \{nombre\}$               | 14. $\{ISBN, DNI, euros\} \rightarrow \{título\}$         |
| 5. $\{DNI, ISBN, título\} \rightarrow \{nombre\}$        | 15. $\{ISBN, nombre, euros\} \rightarrow \{título\}$      |
| 6. $\{DNI, ISBN, euros\} \rightarrow \{nombre\}$         | 16. $\{ISBN, DNI, nombre, euros\} \rightarrow \{título\}$ |
| 7. $\{DNI, título, euros\} \rightarrow \{nombre\}$       | 17. $\{DNI, ISBN\} \rightarrow \{euros\}$                 |
| 8. $\{DNI, ISBN, título, euros\} \rightarrow \{nombre\}$ | 18. $\{DNI, ISBN, nombre\} \rightarrow \{euros\}$         |
| 9. $\{ISBN\} \rightarrow \{título\}$                     | 19. $\{DNI, ISBN, título\} \rightarrow \{euros\}$         |
| 10. $\{ISBN, DNI\} \rightarrow \{título\}$               | 20. $\{DNI, ISBN, nombre, título\} \rightarrow \{euros\}$ |

# Ejemplo

`Ha_escrito(DNI, nombre, ISBN, título, euros)`

1.  **$\{DNI\} \rightarrow \{nombre\}$**

2.  ~~$\{DNI, ISBN\} \rightarrow \{nombre\}$~~

3.  ~~$\{DNI, título\} \rightarrow \{nombre\}$~~

4.  ~~$\{DNI, euros\} \rightarrow \{nombre\}$~~

5.  ~~$\{DNI, ISBN, título\} \rightarrow \{nombre\}$~~

6.  ~~$\{DNI, ISBN, euros\} \rightarrow \{nombre\}$~~

7.  ~~$\{DNI, título, euros\} \rightarrow \{nombre\}$~~

8.  ~~$\{DNI, ISBN, título, euros\} \rightarrow \{nombre\}$~~

9.  **$\{ISBN\} \rightarrow \{título\}$**

10.  ~~$\{ISBN, DNI\} \rightarrow \{título\}$~~

11.  ~~$\{ISBN, nombre\} \rightarrow \{título\}$~~

12.  ~~$\{ISBN, euros\} \rightarrow \{título\}$~~

13.  ~~$\{ISBN, DNI, nombre\} \rightarrow \{título\}$~~

14.  ~~$\{ISBN, DNI, euros\} \rightarrow \{título\}$~~

15.  ~~$\{ISBN, nombre, euros\} \rightarrow \{título\}$~~

16.  ~~$\{ISBN, DNI, nombre, euros\} \rightarrow \{título\}$~~

17.  **$\{DNI, ISBN\} \rightarrow \{euros\}$**

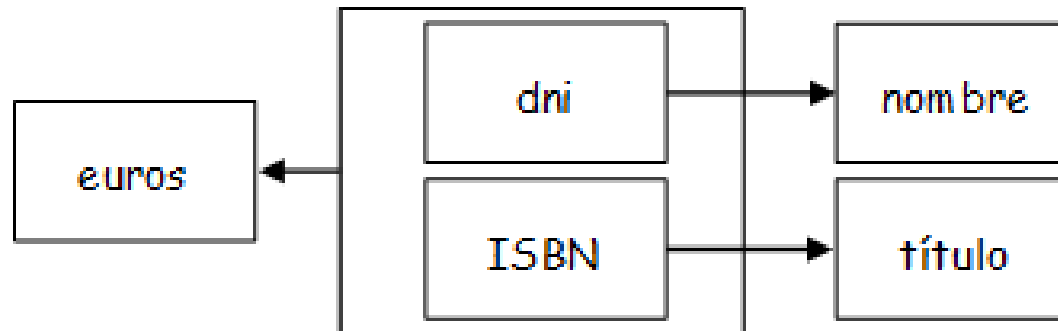
18.  ~~$\{DNI, ISBN, nombre\} \rightarrow \{euros\}$~~

19.  ~~$\{DNI, ISBN, título\} \rightarrow \{euros\}$~~

20.  ~~$\{DNI, ISBN, nombre, título\} \rightarrow \{euros\}$~~

# Ejemplo

Ha\_escrito(DNI, nombre, ISBN, título, euros)



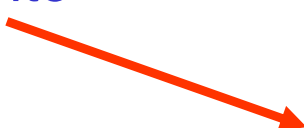
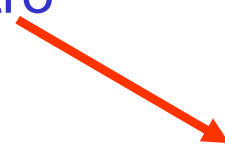
Clave de la relación: {DNI,ISBN}

Atributos primos: DNI e ISBN.

## 5.2 Primera Forma Normal (1FN)

- Una Relación está en 1FN si sus atributos solamente toman valores atómicos (simples e indivisibles).
  - Problemas de utilizar relaciones que no están en 1FN: Hay que utilizar operadores asociados a los tipos de datos complejos (listas, conjuntos, registros,...)

CP: {vcod}

Conjunto 
Registro 

vcod	nombre	teléfonos	dir
V1	Pepe	(963 233 258, 964 523 844, 979 568 987, 987 456 123)	Paz 7, Valencia
V2	Juan	(963 852 741, 910 147 258)	Eolo 3, Castellón
V3	Eva	(987 456 312)	F. Lorca 2, Utiel



## 5.2 Primera Forma Normal (1FN)

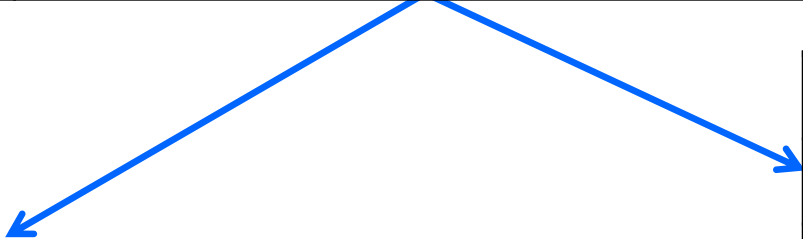
---

- Paso a 1FN:
  - si R tiene un atributo que es un conjunto, eliminarlo de la relación y definir una nueva relación con el atributo plano y buscar la clave primaria.

## 5.2 Primera Forma Normal (1FN)

- Paso a 1ª Forma Normal**

<b>vcod</b>	<b>nombre</b>	<b>teléfonos</b>	<b>dir</b>
V1	Pepe	(963 233 258, 964 523 844, 979 568 987, 987 456 123)	Paz 7, Valencia
V2	Juan	(963 852 741, 910 147 258)	Eolo 3, Castellón
V3	Eva	(987 456 312)	F. Lorca 2, Utiel



<b>vcod</b>	<b>nombre</b>	<b>dir</b>
V1	Pepe	Paz 7, Valencia
V2	Juan	Eolo 3, Castellón
V3	Eva	F. Lorca 2, Utiel

<b>vcod</b>	<b>teléfono</b>
V1	963 233 258
V1	964 523 844
V1	979 568 987
V1	987 456 123
V2	963 852 741
V2	910 147 258
V3	987 456 312

## 5.2 Primera Forma Normal (1FN)

---

Proveedor(vcod, nombre, teléfonos, dir)

CP: {vcod}

Proveedor(vcod, nombre, dir)

CP: {vcod}

Listín(vcod, teléfono)

CP: {teléfono}

CAj: {vcod} → Proveedor

VNN: {vcod}

Si los teléfonos no se  
pueden compartir

## 5.2 Primera Forma Normal (1FN)

---

Proveedor(vcod, nombre, teléfonos, dir)

CP: {vcod}

Proveedor(vcod, nombre, dir)

CP: {vcod}

Listín(vcod, teléfono)

CP: {teléfono,vcod}

CAj: {vcod} → Proveedor

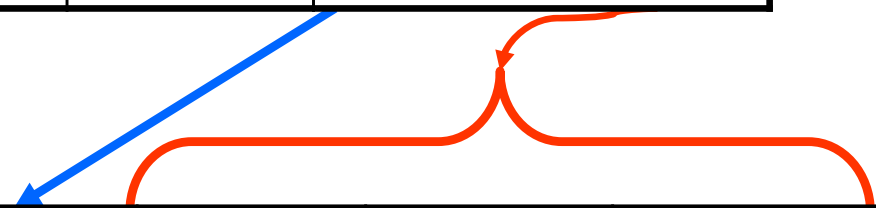
Si los teléfonos se pueden  
compartir

## 5.2 Primera Forma Normal (1FN)

### ➤ Paso a 1FN:

- si R tiene un atributo que es un registro sustituirlo por los campos del registro.

<b>vcod</b>	<b>nombre</b>	<b>dir</b>
V1	Pepe	Paz 7, Valencia
V2	Juan	Eolo 3, Castellón
V3	Eva	F. Lorca 2, Utiel



<b>vcod</b>	<b>nombre</b>	<b>calle</b>	<b>número</b>	<b>ciudad</b>
V1	Pepe	Paz	7	Valencia
V2	Juan	Eolo	3	Castellón
V3	Eva	F. Lorca	2	Utiel

## 5.2 Primera Forma Normal (1FN)

---

Proveedor(vcod, nombre, dir)

CP: {vcod}



Proveedor(vcod, nombre, calle, número, ciudad)

CP: {vcod}

## 5.3 Segunda Forma Normal (2FN)

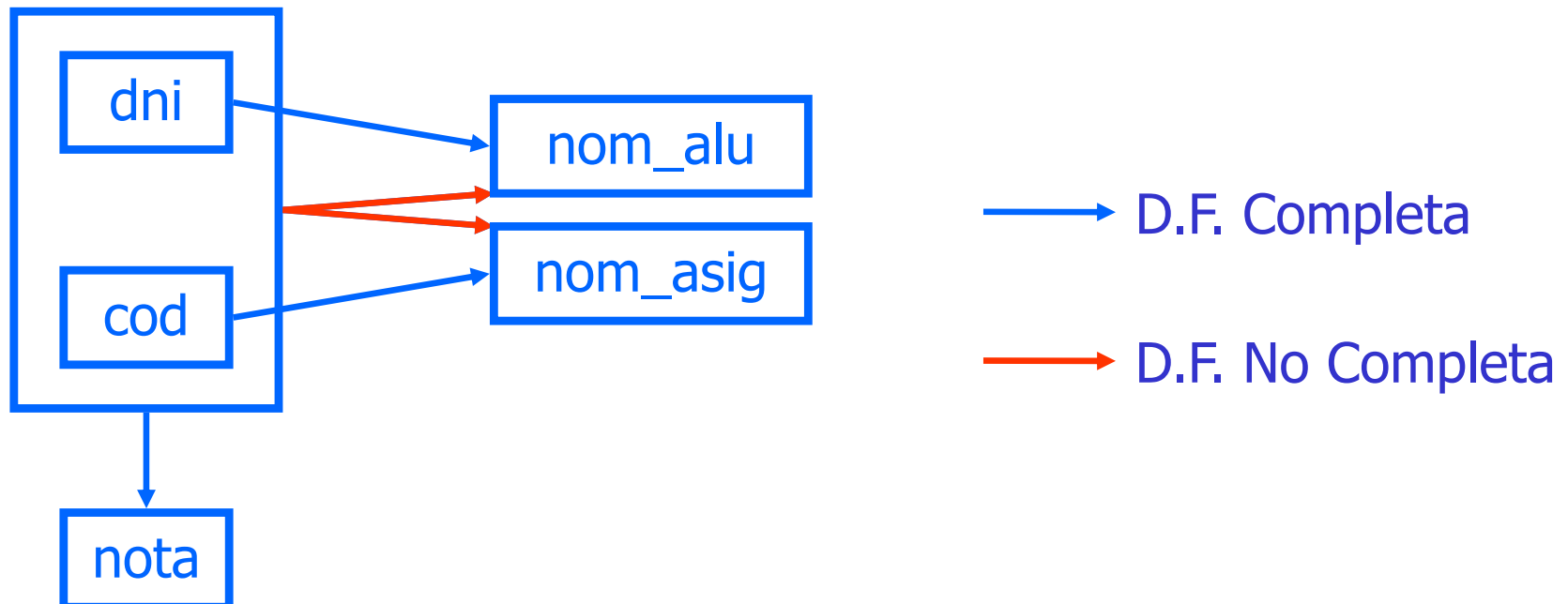
---

- Una Relación está en 2FN si está en 1FN y todo atributo no-primo depende funcionalmente de forma completa de la clave primaria de R.
- Problemas de utilizar relaciones que no están en 2FN
  - Existen redundancias, con lo que se complica la manipulación de la información.
  - No es fácil insertar ni borrar

## 5.2 Segunda Forma Normal (2FN)

CP: {dni,cod}

<b>dni</b>	<b>nom_alu</b>	<b>cod</b>	<b>nom_asig</b>	<b>nota</b>
1	Pepe	DBD	Diseño de Bases de Datos	6
1	Pepe	BDA	Bases de Datos	7
2	Juana	DBD	Diseño de Bases de Datos	7
2	Juana	BDA	Bases de Datos	5





## 5.2 Segunda Forma Normal (2FN)

---

- Paso a 2FN:
  - La clave consta de más de un atributo y existe algún atributo no-primo que no depende completamente de la clave principal. Dividir la relación original en relaciones para eliminar esas dependencias no completas.

## 5.2 Segunda Forma Normal (2FN)

- Paso a 2ª Forma Normal**

dni	nom_alu	cod	nom_asig	nota
1	Pepe	DBD	Diseño de Bases de Datos	6
1	Pepe	BDA	Bases de Datos	7
2	Juana	DBD	Diseño de Bases de Datos	7
2	Juana	BDA	Bases de Datos	5

dni	nom_alu
1	Pepe
2	Juana

cod	nom_asg
DBD	Diseño de Bases de Datos
BDA	Bases de Datos

dni	cod	nota
1	DBD	6
2	BDA	7
1	DBD	7
2	BDA	5

## 5.2 Segunda Forma Normal (2FN)

---

- **2ª Forma Normal**

Matriculado(dni, cod, nom\_alu, nom\_asig, nota)

CP: {dni,cod}

→ Alumno(dni, nom\_alu)

CP: {dni}

→ Asignatura(cod, nom\_asig)

CP: {cod}

→ Matriculado(dni, cod, nota)

CP: {dni,cod}

CAj: {dni} → Alumno

CAj: {cod} → Asignatura

## 5.4 Tercera Forma Normal (3FN)

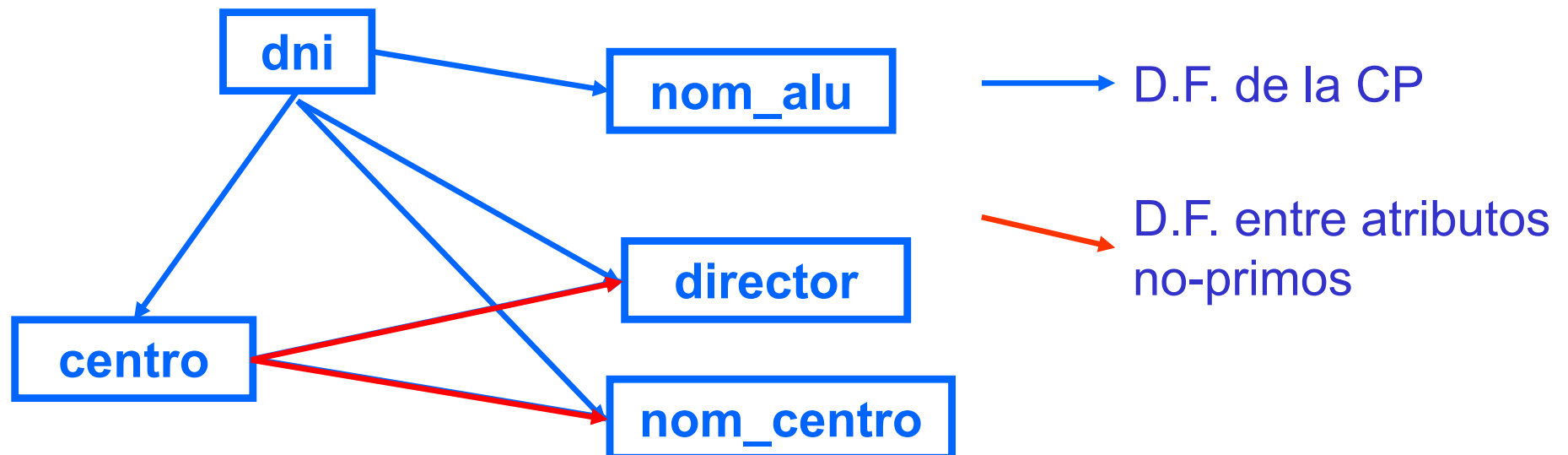
---

- Una Relación está en 3FN si está en 2FN y no hay dependencias funcionales entre atributos no-primos.
- Problemas de utilizar relaciones que no están en 3FN
  - Existen redundancias, con lo que se complica la manipulación de la información.
  - No es fácil insertar ni borrar

## 5.4 Tercera Forma Normal (3FN)

CP: {dni}

<b>dni</b>	<b>nom_alu</b>	<b>centro</b>	<b>nom_centro</b>	<b>director</b>
1	Olga	EUI	Escuela Universitaria de Informática	Pepe
2	Juana	EUI	Escuela Universitaria de Informática	Pepe
3	Ana	FI	Facultad de Informática	Eva
4	Juan	FI	Facultad de Informática	Eva



## 5.4 Tercera Forma Normal (3FN)

---

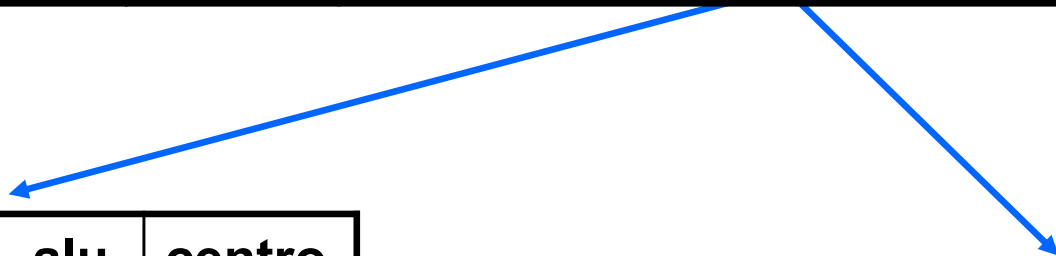
- Paso a 3FN:
  - Existe al menos un par de atributos no-primos que son dependientes.
  - Sacar el atributo dependiente de la relación y crear una nueva cuya clave primaria será el atributo del que depende.

## 5.4 Tercera Forma Normal (3FN)

- Paso a 3ª Forma Normal**

CP: {dni}

<b>dni</b>	<b>nom_alu</b>	<b>centro</b>	<b>nom_centro</b>	<b>director</b>
1	Olga	EUI	Escuela Universitaria de Informática	Pepe
2	Juana	EUI	Escuela Universitaria de Informática	Pepe
3	Ana	FI	Facultad de Informática	Eva
4	Juan	FI	Facultad de Informática	Eva



<b>dni</b>	<b>nom_alu</b>	<b>centro</b>
1	Olga	EUI
2	Juana	EUI
3	Ana	FI
4	Juan	FI

<b>centro</b>	<b>nom_centro</b>	<b>director</b>
EUI	Escuela Universitaria de Informática	Pepe
FI	Facultad de Informática	Eva

## 5.4 Tercera Forma Normal (3FN)

---

Alumno(dni, nom\_alu, centro, nom\_centro, director)

CP: {dni}



Alumno(dni, nom\_alu, centro)

CP: {dni}

CAj: {centro} → Centro\_universitario

Centro\_universitario(centro, nom\_centro, director)

CP: {centro}