# Tema 7b:
# Diseño lógico: normalización

Bases de Datos

Universidad
Rey Juan Carlos

# Índice

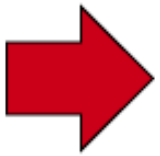
# 1. Introducción

Objetivos:

- Decidir si una relación es "correcta" → no redundante
- Si no, descomponer en {R1, R2, …, Rn}, tal que:
  - Las relaciones sean "correctas"
  - Descomposición "sin pérdidas"

**RELACIÓN**

| Código | Nombre | Ciudad | Provincia |
|---|---|---|---|
| 01 | Juan | Móstoles | Madrid |
| 02 | María | Móstoles | Madrid |
| 03 | Pepe | Navalcarnero | Madrid |
| 04 | Antonio | Valmojado | Toledo |



**EMPLEADOS**

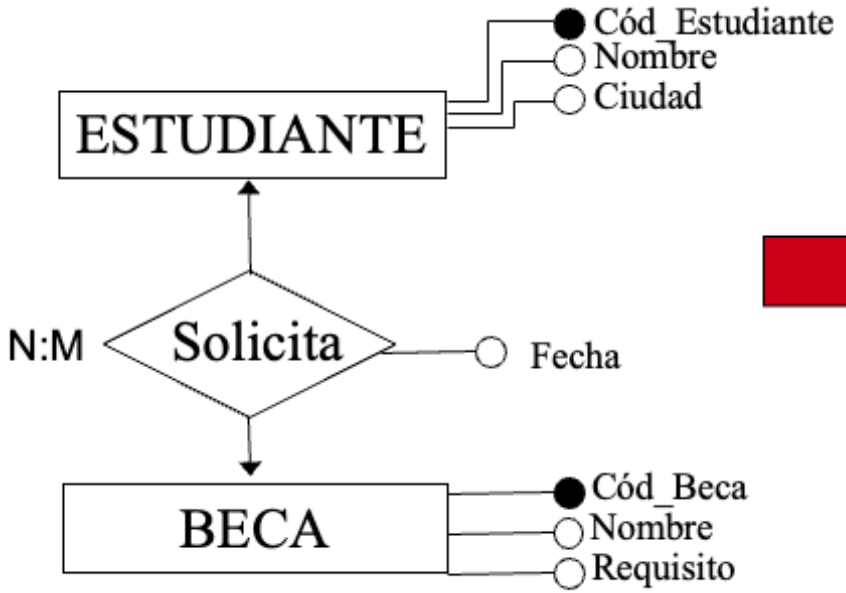
| Código | Nombre | Provincia |
|---|---|---|
| 01 | Juan | Madrid |
| 02 | María | Madrid |
| 03 | Pepe | Madrid |
| 04 | Antonio | Toledo |

**CIUDADES**

| Ciudad | Provincia |
|---|---|
| Móstoles | Madrid |
| Navalcarnero | Madrid |
| Valmojado | Toledo |

*¿Perdemos información con esta descomposición?*
*¿En qué ciudad vive Juan?*

# 1. Introducción




**Metodología de Diseño de BD Relacionales**

¿Es correcto?

ESTUDIANTE (Cód_E, Nombre_E, Ciudad)

SOLICITA (Cód_B, Cód_E, Fecha)

BECA (Cód_B, Nombre_B, Requisito)

# 1. Introducción

**Estudiante**

| Cód_E | Nombre_E | Ciudad |
|---|---|---|
| E1 | Juan | Madrid |
| E2 | María | Madrid |
| E3 | Pepe | Soria |

**Solicita**

| Cód_E | Cód_B | Fecha |
|---|---|---|
| E2 | B2 | 12/11/13 |
| E2 | B1 | 14/10/13 |
| E2 | B3 | 15/09/13 |
| E3 | B3 | 21/09/13 |
| E1 | B2 | 11/11/13 |
| E3 | B2 | 10/10/13 |
| E1 | B1 | 12/11/13 |

**Beca**

| Cód_B | Nombre_B | Requisito |
|---|---|---|
| B1 | Beca1 | IT |
| B2 | Beca2 | IT |
| B3 | Beca3 | II |

**¿Cuál es mejor?**

**Estudiante_Solicita_Beca**

| Cód_E | Nombre_E | Ciudad | Cód_B | Nombre_B | Requisito | Fecha |
|---|---|---|---|---|---|---|
| E2 | **María** | **Madrid** | B2 | Beca2 | IT | 12/11/13 |
| E2 | **María** | **Madrid** | B1 | Beca1 | IT | 14/10/13 |
| E2 | **María** | **Madrid** | B3 | Beca3 | II | 15/09/13 |
| E3 | Pepe | Soria | B3 | Beca3 | II | 21/09/13 |
| E1 | Juan | Madrid | B2 | Beca2 | IT | 11/11/13 |
| E3 | Pepe | Soria | B2 | Beca2 | IT | 10/10/13 |
| E1 | Juan | Madrid | B1 | Beca1 | IT | 12/11/13 |

# 1. Introducción

- Al diseñar una BD relacional, podemos obtener diferentes esquemas.
- La teoría de la normalización consigue una **formalización en el diseño lógico**.
- Representación adecuada de la realidad y problemas de diseños inadecuados.
- La teoría de la normalización permite afrontar el problema de diseño de bases de datos relacionales de una **manera rigurosa y objetiva**.

La normalización se basa en el concepto de Dependencia Funcional.

# 1. Introducción

- Formas de abordar el proceso de modelado:
  A. Obteniendo el esquema relacional directamente
  B. Realizando el proceso de diseño en dos fases (ER y transformación al relacional)

- Posibles problemas (A y B):
  - Incapacidad para almacenar ciertos hechos.
  - Redundancias y, por tanto, posibilidad de inconsistencias
  - Ambigüedades
  - Pérdida de información (aparición de tuplas espurias)
  - Pérdida de dependencias funcionales
  - Existencia de valores nulos (inaplicables)
  - Aparición de estados que no son válidos en el mundo real

# 1. Introducción

Problemas que puede presentar un diseño de un esquema relacional

**Estudiante_Solicita_Beca**

| Cód_E | Nombre_E | Ciudad | Cód_B | Nombre_B | Requisito | Fecha_Solicitud |
|---|---|---|---|---|---|---|
| E2 | María | Madrid | B2 | Beca2 | IT | 12/11/13 |
| E2 | María | Madrid | B1 | Beca1 | IT | 14/10/13 |
| E2 | María | Madrid | B3 | Beca3 | II | 15/09/13 |
| E3 | Pepe | Soria | B3 | Beca3 | II | 21/09/13 |
| E1 | Juan | Madrid | B2 | Beca2 | IT | 11/11/13 |
| E3 | Pepe | Soria | B2 | Beca2 | IT | 10/10/13 |
| E1 | Juan | Madrid | B1 | Beca1 | IT | 12/11/13 |

- ¿Diseño "correcto"?
- Anomalías:
  A. Inserción -> p.e. Estudiantes sin beca no se pueden introducir
  B. Borrado -> p.e. Borrado de solicitud de beca -> borrado de alumno
  C. Modificación -> p.e. Ciudad (inconsistencias)

# 1. Introducción

Problemas que puede presentar un diseño inadecuado de un esquema relacional:

**LIBRO (código, título, nombre_e, dirección_e)**

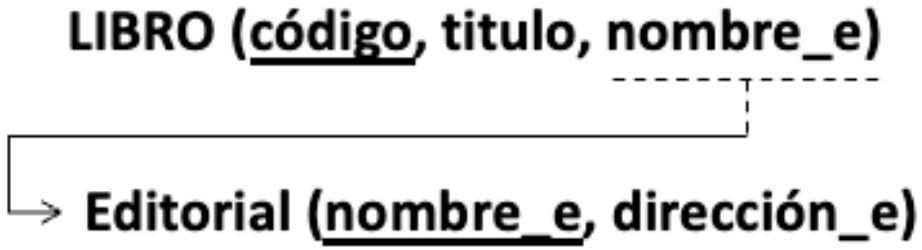
- Anomalías de inserción:
  A. Si no se conoce el nombre de la editorial no se puede insertar un libro.
  B. Al dar de alta una editorial obligaría a insertar tantas tuplas como libros tenga.

- Anomalías de borrado:
  A. Al borrar un libro, si este es el único de una editorial, pierdo la dirección de ésta.
  B. Para borrar una editorial, hay que borrar tantas tuplas como libros tenga.

- Anomalías de modificación:
  A. Para modificar la dirección de la editorial hay que hacerlo en tantas tuplas como libros tenga.
  B. Puede ocurrir que una tupla tenga una dirección para la editorial y otra tupla tenga una dirección distinta para la misma editorial.

# 1. Introducción

Posible solución: Esquema relacional normalizado




- ¿Anomalías de inserción?
  A. Si no se conoce el nombre de la editorial no se puede insertar un libro.
  B. Al dar de alta una editorial obligaría a insertar tantas tuplas como libros tenga.

- ¿Anomalías de borrado?
  A. Al borrar un libro, si este es el único de una editorial, pierdo la dirección de ésta.
  B. Para borrar una editorial, hay que borrar tantas tuplas como libros tenga.

- ¿Anomalías de modificación?
  A. Para modificar la dirección de la editorial hay que hacerlo en tantas tuplas como libros tenga.
  B. Puede ocurrir que una tupla tenga una dirección para la editorial y otra tupla tenga una dirección distinta para la misma editorial.

# 2. Dependencias funcionales

- Dependencias son **propiedades inherentes** al contenido semántico de los datos.
- Forman parte de las **restricciones de usuario** del Modelo Relacional.
- Se han de cumplir en **cualquier extensión** de un esquema de relación (son invariantes en el tiempo).
- **No es posible deducir una dependencia** a partir de la **observación de una extensión** del esquema de relación.
- Existen **diferentes tipos de dependencias**: Funcionales, multivaluadas, jerárquicas y de combinación.
- Por simplicidad: *Para estudio de las dependencias consideramos que el esquema relacional está compuesto por un único esquema de relación:*

  R (A, DEP)
  - A: conjunto de atributos de la relación R
  - DEP: conjunto de dependencias existentes entre los atributos
    - Dependencia: A -> B

# 2. Dependencias funcionales

- Dependencias entre los datos

**NOTAS**

| MAT | Nombre | Apellido | Población | Provincia | Asignatura | Conv | Nota |
|---|---|---|---|---|---|---|---|
| 001 | Juan | Pérez | Móstoles | Madrid | DBD | Sep/09 | NP |
| 001 | Juan | Pérez | Móstoles | Madrid | DBD | Jun/10 | SB |
| 002 | María | García | Alcorcón | Madrid | Redes | Jun/10 | AP |
| 003 | Eva | Gómez | Madrid | Madrid | DBD | Sep/09 | NOT |
| 003 | Eva | Gómez | Madrid | Madrid | Redes | Jun/10 | NOT |
| 003 | Eva | Gómez | Madrid | Madrid | SSOO | Jun/10 | NOT |
| 004 | Luis | Martín | Valmojado | Toledo | DBD | Sep/09 | SUSP |

MAT → Nombre, Apellido, Población, Provincia

Población → Provincia (no porque hay poblaciones iguales para distintas provincias)

MAT+Asignatura+Conv → Nota

¿Nota → Provincia? ¿casualidad?

Las dependencias se deben cumplir SIEMPRE.

# 2. Dependencias funcionales

- Dependencias entre los datos:
  - Semánticas: se pueden recoger con la clave primaria

**ESTUDIANTE**

| MAT | Nombre | Apellido | Población | Provincia |
|---|---|---|---|---|
| 001 | Juan | Pérez | Móstoles | Madrid |
| 002 | María | García | Alcorcón | Madrid |
| 003 | Eva | Gómez | Madrid | Madrid |
| 004 | Luis | Martín | Valmojado | Toledo |

  - Si no conocemos el dominio:

    ¿A -> B?, ¿B -> C?, ¿C -> B?

**T1**

| A | B | C | D | E |
|---|---|---|---|---|
| 001 | B1 | 1 | 1254 | X |
| 002 | B2 | 7 | 5478 | Y |
| 003 | B3 | 54 | 2568 | Z |
| 004 | B4 | 0 | 3698 | F |
| 005 | B3 | 54 | 4758 | X |
| 006 | B6 | 21 | 2678 | S |
| 007 | B3 | 54 | 2147 | S |
| 008 | B2 | 7 | 2147 | T |

# 2. Dependencias funcionales

- Dependencias entre los datos:

¿A -> B?, ¿B -> C?, ¡C-/>B!

**T1**

| A | B | C | D | E |
|---|---|---|---|---|
| 001 | B1 | 1 | 1254 | X |
| 002 | B2 | 7 | 5478 | Y |
| 003 | B3 | 54 | 2568 | Z |
| 004 | B4 | 0 | 3698 | F |
| 005 | B3 | 54 | 4758 | X |
| 006 | B6 | 21 | 2678 | S |
| 007 | B3 | 54 | 2147 | S |
| 008 | B2 | 7 | 2147 | T |
| 009 | B8 | 7 | 4789 | Q |

**No es posible deducir una dependencia** a partir de la **observación de una extensión** del esquema de relación.

# 2. Dependencias funcionales

Definición:

- Sea el esquema de relación R (A,DF), y sean X, Y dos subconjuntos de A, a los que llamamos **descriptores**.

  Se dice que **Y depende funcionalmente de X** o que **X implica o determina a Y**, y se denota como X → Y si, y sólo si, a cada valor x del descriptor X le corresponde un único valor y del descriptor Y.

- Un **determinante** o **implicante** es un conjunto de atributos del que depende funcionalmente otro conjunto de atributos al que llamamos **implicado**.

- Por ejemplo: Cod_Estudiante → Nombre

  Cod_Estudiante es el implicante y Nombre es el implicado

# 2. Dependencias funcionales

ESTUDIANTE_SOLICITA_BECA

| Cód_E | Nombre_E | Ciudad | Cód_B | Nombre_B | Requisito | Fecha |
|---|---|---|---|---|---|---|
| E2 | María | Madrid | B2 | Beca2 | IT | 12/11/13 |
| E2 | María | Madrid | B1 | Beca1 | IT | 14/10/13 |
| E2 | María | Madrid | B3 | Beca3 | II | 15/09/13 |
| E3 | Pepe | Soria | B3 | Beca3 | II | 21/09/13 |
| E1 | Juan | Madrid | B2 | Beca2 | IT | 11/11/13 |
| E3 | Pepe | Soria | B2 | Beca2 | IT | 10/10/13 |
| E1 | Juan | Madrid | B1 | Beca1 | IT | 12/11/13 |

- En esta extensión:
  - Cód_E → _Nombre_E
  - Cód_E, Cód_B → Fecha
  - Fecha → Ciudad
  - …
  - Y, ¿para toda extensión válida?

DF se refiere al esquema de relación y debe cumplirse siempre

# 2. Dependencias funcionales

Estudiante

| Cód_E | Nombre_E | Ciudad | País |
|---|---|---|---|
| E1 | Juan | Madrid | España |
| E2 | María | Madrid | España |
| E3 | Pepe | Soria | España |

- Si Cód_E es clave, se satisface:
  - Cód_E → Cód_E, Nombre_E, Ciudad, País
- Si A → B y A **no es una clave candidata,** involucrará cierta redundancia (pe. Ciudad → País)
- ¿Interesa reducir el número de dependencias?
  - Hay que hacer que se cumplan las dependencias (SGBD)

# 2. Dependencias funcionales

**Dos descriptores X e Y son <u>equivalentes</u> si:**

- $X \rightarrow Y \cap Y \rightarrow X$, o bien, $X \leftrightarrow Y$
- Ejemplo: Cod_Estudiante $\leftrightarrow$ DNI

**Dependencia funcional X $\rightarrow$ Y es trivial si Y X**

- Ejemplo, Cod_Curso, Cod_Edicion $\rightarrow$ Cod_Curso

**Dependencia funcional plena o completa $\Rightarrow$ Y**

Si $\nexists$ un X' sub X | X' $\rightarrow$ Y

Cod_Curso, Cód_Edición, Cod_Estudieante $\Rightarrow$ Nota

Cod_Estudiante, Cód_Curso => Programa, ya que Cód_Curso $\Rightarrow$ Programa

Cod_Estudiante es un atributo redundante, ajeno o extraño

Atributo extraño (X-X')

# 2. Dependencias funcionales

**Dependencia funcional transitiva:**

- Sea R (X, Y, Z) con:
  - $X \rightarrow Y$, $Y \rightarrow Z$, $Y \not\rightarrow X$ (es decir, X e Y no son equivalentes)
  - Z tiene una dependencia trasitiva respecto a X a través de Y
  - $X \dashrightarrow Z$
  - Si $Z \not\rightarrow Y$, la dependencia transitiva es **estricta**

  - **Ejemplo**
    - DIRECCION (CP, localidad, provincia), con:
    - $CP \rightarrow$ Localidad; Localidad $\rightarrow$ Provincia
    - Localidad $\not\rightarrow$ CP (no son equivalentes)
    - Entonces, CP $\rightarrow$ Provincia, se da una dependencia funcional transitiva
    - Por tanto, CP $\dashrightarrow$ Provincia
    - Además, como Provincia $\not\rightarrow$ Localidad, se da una **dependencia transitiva estricta**.

# 2. Dependencias funcionales

**Cierre de un conjunto de dependencias: $DF^+$**

- $DF^+$: conjunto de todas las dependencias funcionales implicadas por (o deducibles de) DF ($DF \subseteq DF^+$).
- Primera aproximación: **axiomas de Armstrong (**reglas de derivación de dependencias).
- Axiomas básicos:
  - Reflexividad: si $Y \subseteq X$, entonces $X \rightarrow Y$ ($X \rightarrow Y$ trivial)
  - Aumento: si $X \rightarrow Y$, entonces $XZ \rightarrow YZ$
  - Transitividad: si $X \rightarrow Y$ y $Y \rightarrow Z$, entonces $X \rightarrow Z$
- **Se puede demostrar que toda dependencia $X \rightarrow Y$ derivada de DF mediante la aplicación de los axiomas está en $DF^+$.**

# 2. Dependencias funcionales

- Podemos calcular el cierre de un conjunto de dependencias funcionales **$DF^+$** aplicando los axiomas de Armstrong.

**Queremos saber si una DF es cierta ($\in DF^+$):**

- Calcular el cierre de un subconjunto, tal que X sea implicante: **cierre transitivo de un descriptor X** de R respecto al conjunto de dependencias DF, ($X^+_{DF}$)
- Dado R(A, DF), se define $X^+_{DF}$ como un subconjunto de los atributos de A tales que $X \rightarrow X^+_{DF} \in DF^+$, siendo $X^+_{DF}$ máximo en el sentido de que la adición de cualquier atributo vulneraría la condición anterior.
- Consecuencias: con $X^+_{DF}$ es posible el **cálculo de SK y K**
- **Algoritmo de Ullman** (algoritmo del cierre de un descriptor): Transitividad

# 2. Dependencias funcionales

**Algoritmo de Cierre de un Descriptor (Algoritmo de Ullman): Ejemplo**

R ({A, B, C, D, E, F}, DF), donde

DF={A → B, B → A, C → A, D → C, (E, C) → D, A → F, C → F}

- Hallar el cierre del descriptor (E, C)
  - E, C → E, C
  - E, C → E, C, D
  - E, C → E, C, D, A
  - E, C → E, C, D, A, B
  - E, C → E, C, D, A, B, F
  - Por tanto,
    - $(E, C)^+ = \{E, C, D, A, B, F\}$
    - **(E,C) es superclave.**

- Hallar el cierre del descriptor B.
  - Nos interesa saber si una dependencia es cierta
  - Por ejemplo, ¿B → F?
  - A → B y B → A (son equivalentes) por lo tanto B → F

# 2. Dependencias funcionales

**Algoritmo de Cierre de un Descriptor: $X_L^+$ donde L es un conjunto de dependencias**

1. $X^0$=X, en el primer paso el cierre comprende el descriptor
2. $X^{i+1}$= $X^i$ u Y siempre que $Z \rightarrow Y$, $Z \subseteq X^i$
3. $X^{i+1}$= $X^i$ ; ya no se puede incluir ningún atributo más

**Ejemplo:**

**R ({A, B, C, D, E, F}, DF),** queremos hallar el cierre de EC
donde DF= {$A \rightarrow B$, $B \rightarrow A$, $C \rightarrow A$, $D \rightarrow C$, $EC \rightarrow D$, $A \rightarrow F$, $C \rightarrow F$}

1. $EC^0$=EC
2. $EC^1$=ECD por $EC \rightarrow D$ y $EC \in EC^0$
3. $EC^2$=ECDC por $D \rightarrow C$ y $D \in EC^1$
4. $EC^3$=ECDCA ($C \rightarrow A$)
5. $EC^4$=ECDCAB ($A \rightarrow B$)
6. $EC^5$=ECDCABF ($A \rightarrow F$)
7. $EC^6$=$EC^+$

**Ejercicio:** Dado el siguiente esquema de relación:

- R(A,B,C,D, E; A-> B, C->D, D->E)
- Calcular AC+

# 2. Dependencias funcionales

**Superclave SK y Clave candidata K de una relación**

Dado un esquema de relación R(A, DF):

Se denomina superclave SK de la relación R a un subconjunto no vacío de A, tal que SK $\rightarrow$ A, es una consecuencia lógica de DF, siendo, por tanto , un elemento de su cierre:

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

K es una **clave candidata** de la relación R si, además de ser una superclave, no existen Ningún subconjunto estricto de K' de K tal que K' implique también a A (condición de **minimidad**)

$$SK \neq \emptyset \wedge SK \rightarrow A \in DF^{+} \neg \wedge \exists\, K' \subset K \mid K' \rightarrow A$$

# 2. Dependencias funcionales

**Determinación de si un descriptor X es clave de una relación**

Se calcula el cierre de $X^+_{DF}$ de X

- Si X+DF $\not\rightarrow$ A, X no es clave.
- Si $X^+_{DF} \rightarrow$ A, X es una superclave. Si para algún subconjunto $X'_i$ del descriptor, $X'^+_{iDF} \rightarrow$ A, X no es clave (mínimo).
- En caso contrario X es una clave candidata.

**Superclave SK y clave candidata K**

**Ejercicio:**

R ({DNI, Cod_Asignatura, Nota, Telefono}, DF)
DF= { DNI -> Telefono,
DNI, Cod_Asignatura -> Nota}

Calcule las SK y las K. Realizar todos los cierres de descriptores.

# 2. Dependencias funcionales

**Equivalencia de conjuntos de DF**

- **Dado DF, podemos afirmar que:**

$$X \rightarrow Y \in DF^{+} \text{ sii } Y \subseteq X^{+}_{DF}$$

$DF_1$ y $DF_2$ son equivalentes si $DF^{+}_1 = DF^{+}_2$

- Si para toda dependencia $X \rightarrow Y \in DF_2$ se cumple que $Y \subseteq X^{+}_{DF1}$, entonces toda dependencia de $DF_2$ está en $DF_1$ y por tanto $DF_1$ es un recubrimiento de $DF_2$
- Si para toda dependencia $Z \rightarrow W \in DF_1$ se cumple que $W \subseteq Z^{+}_{DF2}$, entonces toda dependencia de $DF_1$ está en $DF_2$ y por tanto $DF_2$ es un recubrimiento de $DF_1$

Si se cumplen ambas condiciones, $DF_1$ y $DF_2$ son mutuamente recubrimientos, luego son equivalentes.

# 2. Dependencias funcionales

**Equivalencia de conjuntos de DF**

**Ejemplo**

DF1={Cod_Curso -> Nombre, Nombre-> Cod_Curso, Cod_Curso -> Cod_Dep, Cod_Curso -> Cod_Programa}

DF2= {Cod_Curso -> Nombre, Nombre-> Cod_Curso, Nombre -> Cod_Dep, Nombre -> Cod_Programa}

- Cod_Curso -> Nombre, Nombre-> Cod_Curso están en las dos

- Cod_Curso + DF2={Cod_Curso, Nombre, Cod_Dep, Cod_Programa}
- Nombre + DF1={Nombre, Cod_Curso, Cod_Dep, Cod_Programa}

DF1 es un recubrimiento de DF2 y DF2 es un recubrimiento de DF1

DF1 y DF2 son equivalentes

# 2. Dependencias funcionales

## Equivalencias redundantes

X->Y es redundante si puede derivarse (deducirse) del conjunto de DF – (X->Y)

Es decir, si $Y \subseteq X^{+}_{DF-(X->Y)}$

### Cálculo del recubrimiento irredundante/minimal

$DF_1$ y $DF_2$ son equivalentes si $DF^{+}_1 = DF^{+}_2$

De todos los posibles conjuntos equivalentes a un conjunto dado de dependencias, hay algunos de ellos que son mínimos, por lo que se dice que son recubrimientos irredundantes (también llamados minimales) del conjunto dado de dependencias:

- Un único atributo en el implicado
- Irreducible por la izquierda (no atributos extraños)
- Sin DF redundantes

# 2. Dependencias funcionales

## Recubrimiento mínimo de un conjunto de DF

Ejercicio:

Sea la relación

ALUMNO ({Num_Alum,DNI,Cod_Asig, Cod_Titulac}

{Num_Alum → DNI,

DNI → Num_Alum,

DNI → Cod_Asig, Cod_Titulac,

Num_Alum → Cod_Asig,

Cod_Asig → Cod_Titulac}

- Un único atributo en el implicado
- Irreducible por la izquierda (no atributos extraños)
- Sin DF redundantes

# 2. Dependencias funcionales

## Algoritmo para la obtención de las calves candidatas de una relación

Partimos de una relación R (A, DF), siendo DF un recubrimiento irredundante:

- Paso 1. Eliminación de descriptores independientes
- Paso 2. Eliminación de descriptores equivalentes
- Paso 3. Determinación de un descriptor (en el que no haya implicados) que sea clave de $R_{sie}$
- Paso 4. Determinación de un descriptor de $R_{sie}$ (en el que puede haber implicados siempre que sean también implicantes)
- Paso 5. Tratamiento de atributos independientes para obtener una clave de la relación original
- Paso 6. Tratamiento de descriptores equivalentes

# 2. Dependencias funcionales

Ejercicio

Recubrimiento mínimo y obtención de claves candidatas

**a)** R ({A,B,C,D},
{A,B→C,
B→D,
B→A,
D→A})

**b)** R ({A,B,C,D},
{A→B,
B,C,D→A})

**c)** R ({A, B, C, D, E, F},
{A → B,
B → C,
C, D → E })

## Recubrimiento Mínimo

- Un único atributo en el implicado
- Irreducible por la izquierda (no atributos extraños)
- Sin DF redundantes

## Algoritmo de Obtención de Claves

**Paso 1.** Eliminación de descriptores independientes
**Paso 2.** Eliminación de descriptores equivalentes
**Paso 3.** Determinación de un descriptor clave $R_{sie}$
**Paso 4.** Determinación de un descriptor de $R_{sie}$ (implicado e implicante)
**Paso 5.** Tratamiento de atributos independientes
**Paso 6.** Tratamiento de descriptores equivalentes

# 2. Dependencias funcionales

Ejercicio (obtención de claves candidatas)

R ({A,B,C,D,E,F, G, H, I, J}

AB → C, C → >A, C → B, E → DF, D → E, F → F, ABD → G, CF → H )

Pasos a seguir:

1. Obtener recubrimiento minimal de DF
   - Un implicado en cada dependencia
   - Dependencias sin atributos extraños

     X es extraño en XY → Z si Y → Z Є DF+ <=> Z Є Y+DF
   - Sin dependencias redundantes

     X → Y es redundante si Y ⊆ X + (DF – X → Y)

# 2. Dependencias funcionales

Ejercicio (obtención de claves candidatas) II

R ({A,B,C,D,E,F, G, H, I, J}

AB → C, C → A, C → B, E → DF, D → E, F → F, ABD → G, CF → H )

Pasos a seguir:

2. Obtener claves
    - Paso 1. Eliminación de descriptores independientes
    - Paso 2. Eliminación de descriptores equivalentes
    - Paso 3. Determinación de un descriptor (en el que no haya implicados) que sea clave de $_{Rsie}$
    - Paso 4. Determinación de un descriptor de $R_{sie}$ (en el que puede haber implicados siempre que sean también implicantes)
    - Paso 5. Tratamiento de atributos independientes para obtener una clave de la relación original
    - Paso 6. Tratamiento de descriptores equivalentes

**Atributo Principal**: aquel que aparece en alguna clave K

**Atributo No Principal**: aquel que no aparece en ninguna clave K

# 3. Teoría de la normalización

Dado un esquema de relación R(A,DF), donde:

- A es un conjunto de atributos
- DF es el conjunto de dependencias existentes entre ellos

Se trata de **transformar**, por medio de sucesivas proyecciones, este esquema de partida en un conjunto de n esquemas de relación:

- {Ri (Ai, DFi)}n i=1, tales que cumplan unas determinadas condiciones:

El conjunto de esquemas Ri deberán ser equivalentes a R y mejores que el esquema de partida

# 3. Teoría de la normalización

Descomposición sin pérdida de información:

**Estudiante**

| Cód_E | Nombre_E | Ciudad |
|---|---|---|
| E1 | Juan | Madrid |
| E2 | María | Madrid |
| E3 | Pepe | Soria |

**Solicita**

| Cód_E | Cód_B | Fecha |
|---|---|---|
| E2 | B2 | 12/11/13 |
| E2 | B1 | 14/10/13 |
| E2 | B3 | 15/09/13 |
| E3 | B3 | 21/09/13 |
| E1 | B2 | 11/11/13 |
| E3 | B2 | 10/10/13 |
| E1 | B1 | 12/11/13 |

**Beca**

| Cód_B | Nombre_B | Requisito |
|---|---|---|
| B1 | Beca1 | IT |
| B2 | Beca2 | IT |
| B3 | Beca3 | II |

**Estudiante * Solicita * Beca**

| Cód_E | Nombre_E | Ciudad | Cód_B | Nombre_B | Requisito | Fecha |
|---|---|---|---|---|---|---|
| E2 | María | Madrid | B2 | Beca2 | IT | 12/11/13 |
| E2 | María | Madrid | B1 | Beca1 | IT | 14/10/13 |
| E2 | María | Madrid | B3 | Beca3 | II | 15/09/13 |
| E3 | Pepe | Soria | B3 | Beca3 | II | 21/09/13 |
| E1 | Juan | Madrid | B2 | Beca2 | IT | 11/11/13 |
| E3 | Pepe | Soria | B2 | Beca2 | IT | 10/10/13 |
| E1 | Juan | Madrid | B1 | Beca1 | IT | 12/11/13 |

# 3. Teoría de la normalización

Descomposición sin pérdida de información:

**Estudiante_Nueva**

| Cód_E | Nombre_E | Ciudad | Fecha |
|---|---|---|---|
| E2 | María | Madrid | 12/11/13 |
| E2 | María | Madrid | 14/10/13 |
| E2 | María | Madrid | 15/09/13 |
| E3 | Pepe | Soria | 21/09/13 |
| E1 | Juan | Madrid | 11/11/13 |
| E3 | Pepe | Soria | 10/10/13 |
| E1 | Juan | Madrid | 12/11/13 |

**Beca_Nueva**

| Cód_B | Nombre_B | Requisito | Fecha |
|---|---|---|---|
| B2 | Beca2 | IT | 12/11/13 |
| B1 | Beca1 | IT | 14/10/13 |
| B3 | Beca3 | II | 15/09/13 |
| B3 | Beca3 | II | 21/09/13 |
| B2 | Beca2 | IT | 11/11/13 |
| B2 | Beca2 | IT | 10/10/13 |
| B1 | Beca1 | IT | 12/11/13 |

**Estudiante_Nueva * Beca_Nueva**

| Cód_E | Nombre_E | Ciudad | Cód_B | Nombre_B | Requisito | Fecha |
|---|---|---|---|---|---|---|
| E2 | María | Madrid | B2 | Beca2 | IT | 12/11/13 |
| E2 | María | Madrid | B1 | Beca1 | IT | 12/11/13 |
| E2 | María | Madrid | B1 | Beca1 | IT | 14/10/13 |
| E2 | María | Madrid | B3 | Beca3 | II | 15/09/13 |
| E3 | Pepe | Soria | B3 | Beca3 | II | 21/09/13 |
| E1 | Juan | Madrid | B2 | Beca2 | IT | 11/11/13 |
| E3 | Pepe | Soria | B2 | Beca2 | IT | 10/10/13 |
| E1 | Juan | Madrid | B1 | Beca1 | IT | 12/11/13 |
| E1 | Juan | Madrid | B2 | Beca2 | IT | 12/11/13 |

# 3. Teoría de la normalización

Descomposición sin pérdida de información:

**Estudiante_Nueva**

| Cód_E | Nombre_E | Ciudad | Fecha |
|---|---|---|---|
| E2 | María | Madrid | 12/11/13 |
| E2 | María | Madrid | 14/10/13 |
| E2 | María | Madrid | 15/09/13 |
| E3 | Pepe | Soria | 21/09/13 |
| E1 | Juan | Madrid | 11/11/13 |
| E3 | Pepe | Soria | 10/10/13 |
| E1 | Juan | Madrid | 12/11/13 |

**Beca_Nueva**

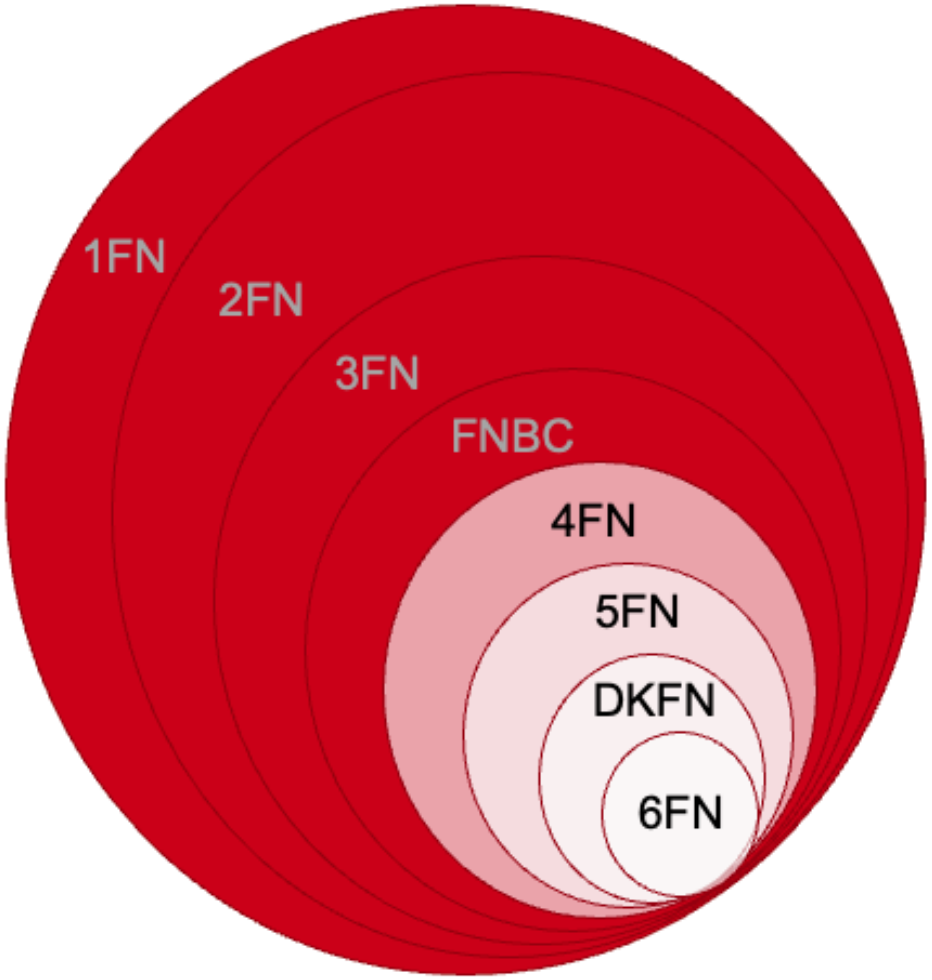
| Cód_B | Nombre_B | Requisito | Fecha |
|---|---|---|---|
| B2 | Beca2 | IT | 12/11/13 |
| B1 | Beca1 | IT | 14/10/13 |
| B3 | Beca3 | II | 15/09/13 |
| B3 | Beca3 | II | 21/09/13 |
| B2 | Beca2 | IT | 11/11/13 |
| B2 | Beca2 | IT | 10/10/13 |
| B1 | Beca1 | IT | 12/11/13 |

**Estudiante_Nueva * Beca_Nueva**

| Cód_E | Nombre_E | Ciudad | Cód_B | Nombre_B | Requisito | Fecha |
|---|---|---|---|---|---|---|
| E2 | María | Madrid | B2 | Beca2 | IT | 12/11/13 |
| E2 | María | Madrid | B1 | Beca1 | IT | 12/11/13 |
| E2 | María | Madrid | B1 | Beca1 | IT | 14/10/13 |
| E2 | María | Madrid | B3 | Beca3 | II | 15/09/13 |
| E3 | Pepe | Soria | B3 | Beca3 | II | 21/09/13 |
| E1 | Juan | Madrid | B2 | Beca2 | IT | 11/11/13 |
| E3 | Pepe | Soria | B2 | Beca2 | IT | 10/10/13 |
| E1 | Juan | Madrid | B1 | Beca1 | IT | 12/11/13 |
| E1 | Juan | Madrid | B2 | Beca2 | IT | 12/11/13 |

# 4. Formas normales

- Un esquema de relación está en **una cierta forma normal** si satisface un conjunto de restricciones.
- Cuanto más alta sea la forma normal **menores serán los problemas en el mantenimiento** de la BD.
- Codd propuso inicialmente tres formas normales basadas en las DF: 1FN, 2FN y 3FN.
- En 1974 introdujo una definición más restrictiva de la 3FN que se denominó Forma Normal de Boyce-Codd (FNBC).
- La 4FN se basa dependencias multivaluadas y la 5FN en las dependencias de proyección-combinación.



# 4. Formas normales

## Primera forma normal

- Una relación está en 1FN cuando cada atributo sólo toma un valor del dominio simple subyacente

**Estudiante (Código, Nombre, Beca)**

| Código | Nombre | Beca |
|---|---|---|
| E1 | Juan | Beca1<br>Beca2 |
| E2 | María | Beca3 |
| E3 | Pepe | Beca2<br>Beca3 |

Tabla, no relación
Grupos repetitivos

| Código | Nombre | Beca |
|---|---|---|
| E1 | Juan | Beca1 |
| E1 | Juan | Beca2 |
| E2 | María | Beca3 |
| E3 | Pepe | Beca2 |
| E3 | Pepe | Beca3 |

1FN

# 4. Formas normales

## Segunda forma normal

**Una relación está en 2FN si:**

- Está en 1FN.
- Cada atributo no principal tiene DF completa respecto de cada una de las claves.

- No se cumple cuando **algún atributo no principal** (atributo que no forme parte de la clave) **depende funcionalmente de algún subconjunto de la clave** (depende parcialmente de la clave)
- Cualquier relación binaria (relación con dos atributos), cualquier relación cuyas claves son simples y cualquier relación en la que todos sus atributos son principales, está en 2FN.
- **Siempre se pueden obtener esquemas en 2FN sin pérdida de información ni de dependencias.**

# 4. Formas normales

## Segunda forma normal

**Ejemplo:**

- ESTUDIANTE_BECA (AT, DEP) donde:
  - AT = {Cod_Estudiante, Cod_Beca, Fecha_Sol, Título}
  - DEP = { Cod_Estudiante, Cod_Beca → Fecha_Sol,
    **Cod_Estudiante → Título**}

K= {Cod_Estudiante, Cod_Beca}

Atributos No Principales= Fecha_Sol, Título

**Título** no es una información acerca de la totalidad de la clave, sino de parte de la misma.
No se encuentra en 2FN

Transformamos en:

- ESTUDIANTE_BECA1 (AT1, DEP1) donde:
  - AT1 = {Cod_Estudiante, Cod_Beca, Fecha_Sol }
  - DEP1 = {Cod_Estudiante, Cod_Beca → Fecha_Sol }
- ESTUDIANTE (AT2, DEP2) donde:
  - AT2 = {Cod_Estudiante, Título}
  - DEP2 = {Cod_Estudiante → Título}

**¿En qué forma normal estarían estas relaciones?**

# 4. Formas normales

## Tercera forma normal

**Una relación está en 3FN si, y sólo si:**

- Está en 2FN.
- No existe ningún **atributo no principal** (que no forme parte de ninguna clave) **que dependa transitivamente de alguna clave de R**. Es decir, todos los determinantes, o bien son superclave, o bien son atributos principales.

- No se cumple cuando existen atributos no principales que dependen funcionalmente de otros atributos no principales.
- Toda relación binaria y toda relación que tiene, como mucho, un atributo no principal, está en 3FN.
- **Siempre se pueden obtener esquemas de relación en 3FN sin pérdida de información ni de DF.**

# 4. Formas normales

## Tercera forma normal

**Ejemplo:**

- ESTUDIANTE (AT, DEP) donde:
  - AT = {Cod_Estudiante, Cod_Proyecto, Nombre_Proyecto}
  - DEP = { Cod_Estudiante → **Cod_Proyecto,**
    **Cod_Proyecto → Nombre_Proyecto**,}

Transformamos en:

K= {Cod_Estudiante}

ANP= Cod_Proyecto, Nombre_Proyecto

**No** está en **3FN**
-> Está en 2FN

- ESTUDIANTE1 (AT1,DEP1) donde:
  - AT1={Cod_Estudiante, Cod_Proyecto}
  - DEP1={Cod_Estudiante → Cod_Proyecto}
- PROYECTO (AT2,DEP2) donde:
  - AT2={Cod_Proyecto, Nombre_Proyecto}
  - DEP2={Cod_Proyecto → Nombre_Proyecto}

# 4. Formas normales

## Forma normal de Boyce-Codd

**Una relación está en FNBC si, y sólo si:**

- Todo implicante (determinante) es una clave candidata.

ASISTE ({Cod_Curso, Nom_Curso, Cod_Estud, Calif}, {Cod_Curso → Nom_Curso,

Cod_Curso, Cod_Estudiante → Calif})

K={Cod_Curso, Cod_Estud}, {Nom_Curso, Cod_Estud}

ANP=Calif y ANP depende de la K => 3FN

PERO: Se repite nombre y código por cada estudiante que asiste al curso, porque Cod_Curso y Nom_Curso son implicantes, pero no son K.

- Si 3FN y claves no solapadas, también en FNBC.
- Toda relación binaria está en FNBC.

**<u>En la transformación pueden perderse DF.</u>**

# 4. Formas normales

**Forma normal de Boyce-Codd**

**Ejemplo:**

- SE_MATRICULA1 ({Cod_Curso, Cod_Edición , Cod_Estudiante, Fecha}, {Cod_Curso,Cod_Edición,Cod_Estudiante→Fecha, Cod_Edición,Cod_Estudiante,Fecha→Cod_Curso})
- Claves candidatas:
  - (Cod_Curso, Cod_Edición, Cod_Estudiante)
  - (Cod_Edición,Cod_Estudiante,Fecha)

Claves Solapadas

- Se solapan ya que comparten los atributos Cod_Edición y Cod_Estudiante; sin embargo, debido a que los únicos determinantes son los dos descriptores anteriores, que son claves candidatas, la relación sí se encuentra en FNBC.

# 4. Formas normales

## Forma normal de Boyce-Codd

No siempre es posible transformar un esquema de relación **que no está en FNBC sin pérdida de DF** (sí se puede asegurar que se realice sin pérdida de información):

- CLASE (AT,DEP) donde:
  - AT={Cod_Estudiante, Cod_Profesor, Materia}
  - DEP={(Cod_Estudiante, Materia) → Cod_Profesor, Cod_Profesor → Materia}
  - Claves candidatas K:
    - {Cod_Estudiante, Materia} y {Cod_Estudiante, Cod_Profesor}
    - Atributos No Principales: todos los atributos son principales
- 3FN, Claves candidatas compuestas que se solapan (El implicante Cod_Profesor no es clave candidata)

CLASE1 (AT1,DEP1) donde:
- AT1={Cod_Estudiante, Cód_Profesor}
- DEP1={}

CLASE2 (AT2,DEP2) donde:
- AT2={Cod_Profesor, Materia}
- DEP2={Cod_Profesor → Materia}

Pérdida de DF (Cod_Estudiante, Materia) → Cod_Profesor

# 4. Formas normales

**Resumen**

- 1FN : sin grupos repetitivos
- 2FN: 1FN y no pueden existir DF tal que **parte de la clave implique a un atributo no principal** (no existencia de atributos extraños)
- 3FN: 2FN y un **atributo no principal** no puede implicar un atributo no principal.
- FNBC: Todas las dependencias deben ser tal que **K → cualquier cosa** (posible problema: claves candidatas compuestas solapadas)

# 5. Proceso de descomposición

**Aspectos a tener en cuenta:**

- Avanzar en las formas normales
- Conservación de los atributos
- Conservación de las dependencias
- Conservación de la información (lossless join)

Existen muchos algoritmos

# 5. Proceso de descomposición

## Conservación de las dependencias

| COD_E | CP | CIUDAD |
|---|---|---|
| E1 | 28008 | Madrid |
| E2 | 28008 | Madrid |

$COD\_E \rightarrow CP$

$CP \rightarrow CIUDAD$

| COD_E | CP |
|---|---|
| E1 | 28008 |
| E2 | 28008 |

$COD\_E \rightarrow CP$

| CP | CIUDAD |
|---|---|
| 28008 | Madrid |
| 28008 | Madrid |

$CP \rightarrow CIUDAD$

| COD_E | CP |
|---|---|
| E1 | 28008 |
| E2 | 28008 |

$COD\_E \rightarrow CP$

| COD_E | CIUDAD |
|---|---|
| E1 | Madrid |
| E2 | Madrid |

¿$CP \rightarrow CIUDAD$?

# 5. Proceso de descomposición

**La descomposición implica separar un esquema en otros, en un proceso reversible.**

| COD_E | PUESTO | CIUDAD |
|---|---|---|
| E1 | Administrativo | Madrid |
| E2 | Administrativo | Móstoles |

| COD_E | PUESTO |
|---|---|
| E1 | Administrativo |
| E2 | Administrativo |

| COD_E | CIUDAD |
|---|---|
| E1 | Madrid |
| E2 | Móstoles |

| COD_E | PUESTO | CIUDAD |
|---|---|---|
| E1 | Administrativo | Madrid |
| E2 | Administrativo | Móstoles |

| COD_E | PUESTO |
|---|---|
| E1 | Administrativo |
| E2 | Administrativo |

| PUESTO | CIUDAD |
|---|---|
| Administrativo | Madrid |
| Administrativo | Móstoles |

| COD_E | PUESTO | CIUDAD |
|---|---|---|
| E1 | Administrativo | Madrid |
| E1 | Administrativo | Móstoles |
| E2 | Administrativo | Madrid |
| E2 | Administrativo | Móstoles |

# 5. Proceso de descomposición

- Comprobación de descomposición sin pérdida de información en caso de descomposiciones binarias.
- Una descomposición D = {R1, R2} es sin pérdida de información (lossless join) con respecto a un conjunto de DF sii

  $(A1 \cap A2) \rightarrow (A1 - A2)$ ó

  $(A1 \cap A2) \rightarrow (A2 - A1)$

**Atributo común a ambas relaciones debe ser clave en alguna de las dos relaciones.**

# 5. Proceso de descomposición

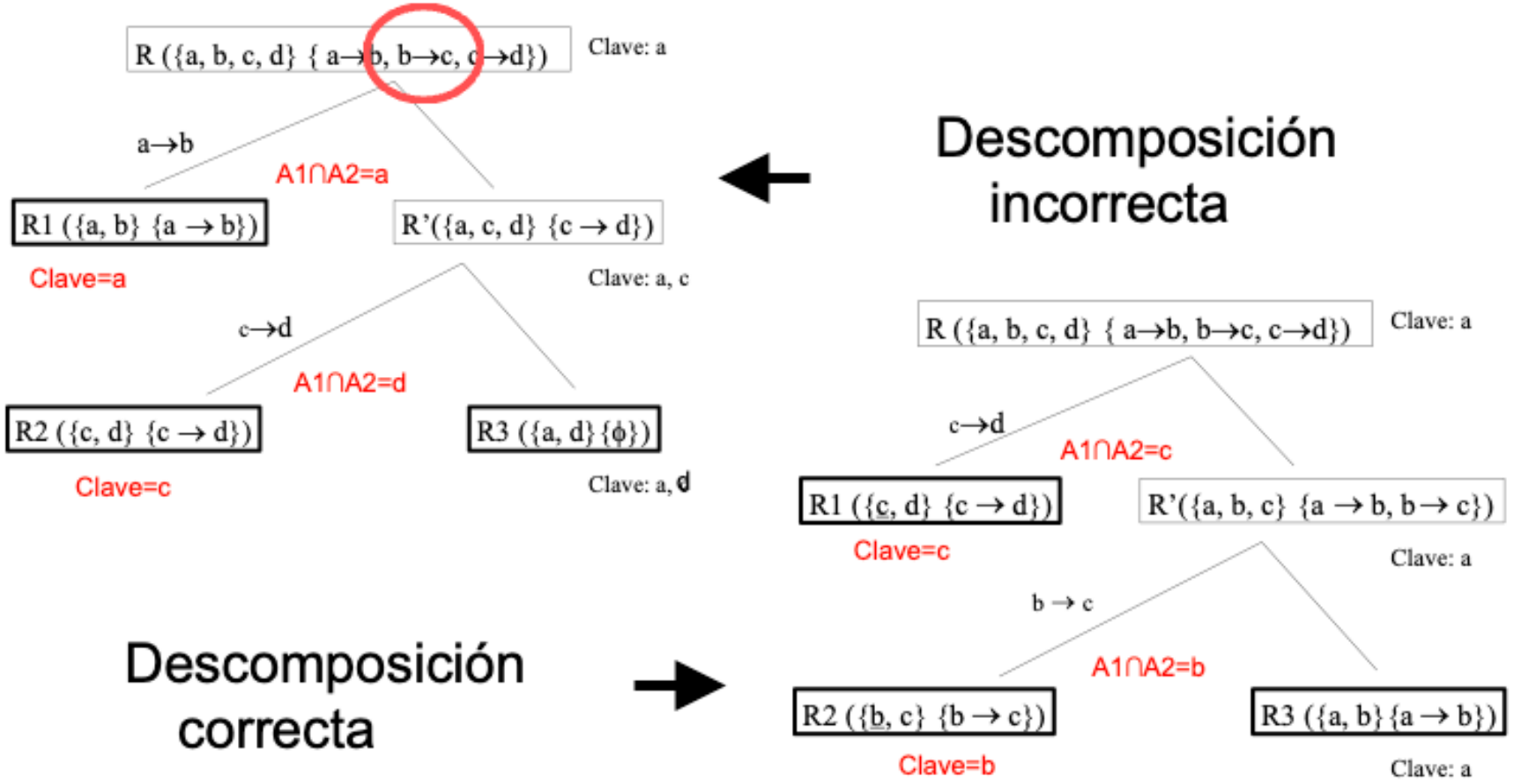
**Algoritmo de descomposición sin pérdida**

1. Hallar un recubrimiento minimal $DF^m$
2. Determinar la(s) clave(s), así como los atributos principales y no principales
3. Identificar la FN en que se encuentra la relación

**Si se desea llegar a una forma normal más avanzada:**

4. Agrupar las DF que tengan el mismo implicante y/o equivalentes.
5. Obtener proyecciones independientes sobre cada una de las DF (o de los grupos), de forma que los atributos que aparecen en la DF constituyen una nueva relación y los atributos no principales, así como la DF, desaparezcan de la relación origen.
6. Proseguir esta descomposición repitiendo el paso 5 hasta que no pueda continuarse porque todas las DF estén implicadas por una clave.

# 5. Proceso de descomposición




4. Agrupar las DF que tengan el **mismo implicante** y/o equivalentes.
5. Obtener **proyecciones independientes** sobre **cada una de las DF** (o de los grupos), de forma que los atributos que aparecen en la DF constituyen una nueva relación y los atributos no principales, así como la DF, desaparezcan de la relación origen.

# Bibliografía

**Básica:**

- **Tecnología y diseño de bases de datos**

  *M. Piattini, E. Marcos, C. Calero y B. Vela.*

  *Ed.: RAMA 2006*

  *Parte IV – Capítulos 9-12*

# Bibliografía

**Complementaria:**

- **An Introduction to Database Systems**

  *Date, C.J.*

  *Ed.: Addison-Wesley, 2024. 8a edición*

- **Diseño Conceptual de Bases de Datos. Un enfoque de entidades-interrelaciones**.

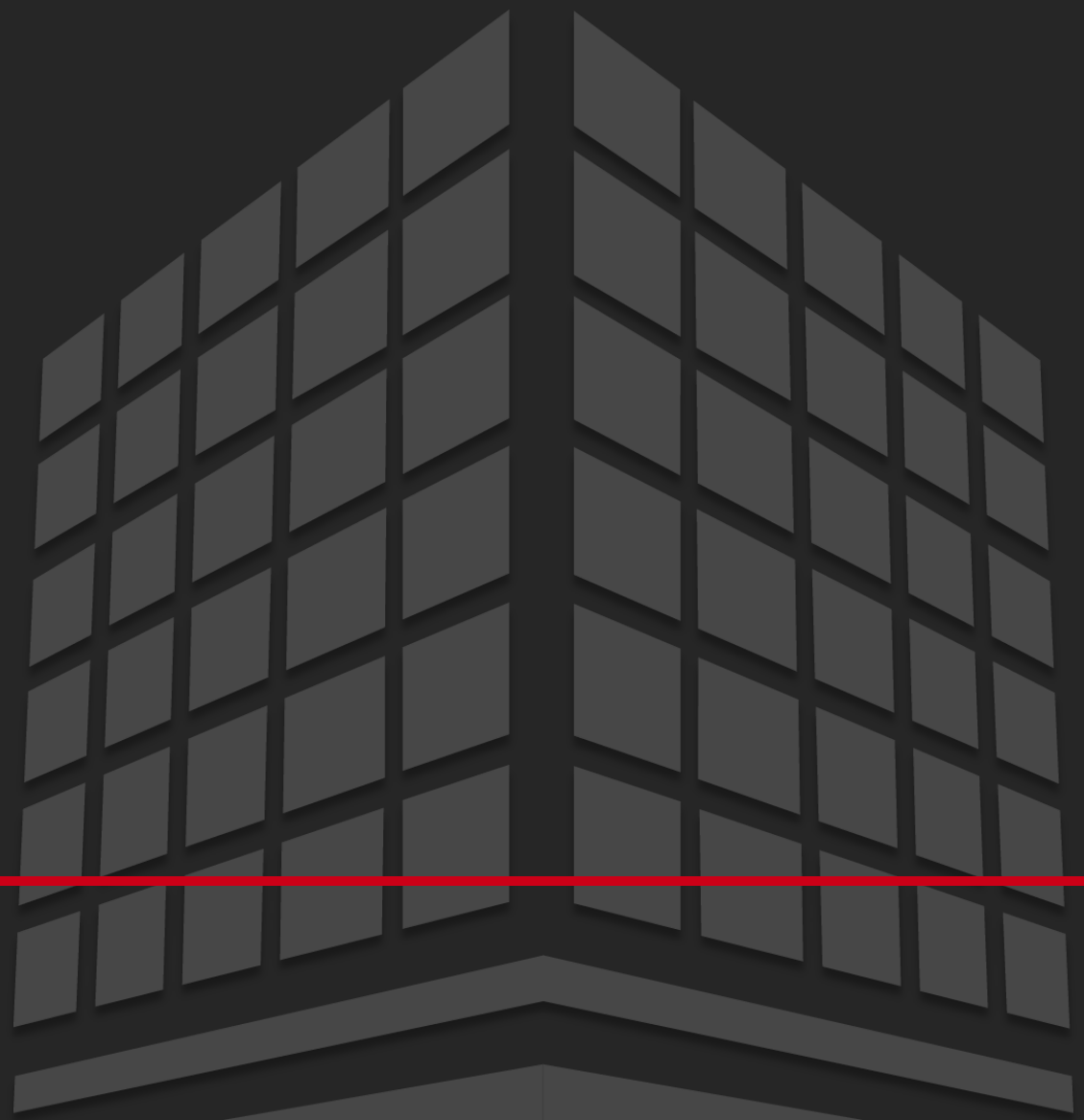
  *Batini, C. Ceri, S., Navathe, S. B*

  *Addison-Wesley Iberoamericana, 1994*

- **Sistemas de Bases de Datos. Conceptos Fundamentales**.

  *Elsmari, R. y Navathe, S. B.*

  *Addison-Wesley Iberoamericana, 1997*



Universidad
Rey Juan Carlos