

8. Teoría de la Normalización

Objetivos

- Apreciar la **importancia** y **utilidad** de emplear la teoría de la **normalización** en la etapa de **diseño lógico** de bases de datos relacionales, para **detectar y corregir esquemas relacionales diseñados inadecuadamente**.
- Comprender los conceptos de **dependencia funcional** y **dependencia multivalorada**.
- Entender el significado de las **formas normales 1FN, 2FN, 3FN, FNBC y 4FN**.
- Comprender los diferentes **procedimientos algorítmicos** de la teoría de la normalización que permiten determinar las claves de, o normalizar, una relación.

8. Teoría de la Normalización

Contenidos

- 8.1. Motivación
- 8.2. Dependencias funcionales
- 8.3. Formas normales
- 8.4. Descomposición de relaciones
- 8.5. Enfoques de diseño relacional: Análisis y Síntesis
- 8.6. Dependencias multivaloradas y cuarta forma normal

8. Teoría de la Normalización

Bibliografía

- [EN 2002] Elmasri, R.; Navathe, S.B.: **Fundamentos de Sistemas de Bases de Datos**. 3ª Edición. Addison-Wesley. (Cap. 14 y 15)
- [CB 2005] Connolly, T.; Begg, C.: **Sistemas de bases de datos**. 4ª Edición. Pearson Addison-Wesley. (Cap. 13 y 14)
- [MPM 1999] De Miguel, A.; Piattini, M.; Marcos, E.: **Diseño de bases de datos relacionales**. Ra-Ma. (Cap. 4, 5 y 6)
- [SKS 2002] Silberschatz, A.; Korth, H; Sudarshan, S.: **Fundamentos de bases de datos**. 4ª Edición. McGraw-Hill. (Cap. 7)
- [EN 1997] Elmasri, R.; Navathe, S.B.: **Sistemas de bases de datos. Conceptos fundamentales**. 2ª Edición. Addison-Wesley Iberoamericana. (Cap. 12 y 13)
- [D 2001] Date, C.J.: **Introducción a los sistemas de bases de datos**. 7ª Edición. Prentice-Hall. (Cap. 10, 11 y 12)

8.1 Motivación

Diseño en el modelo relacional de datos

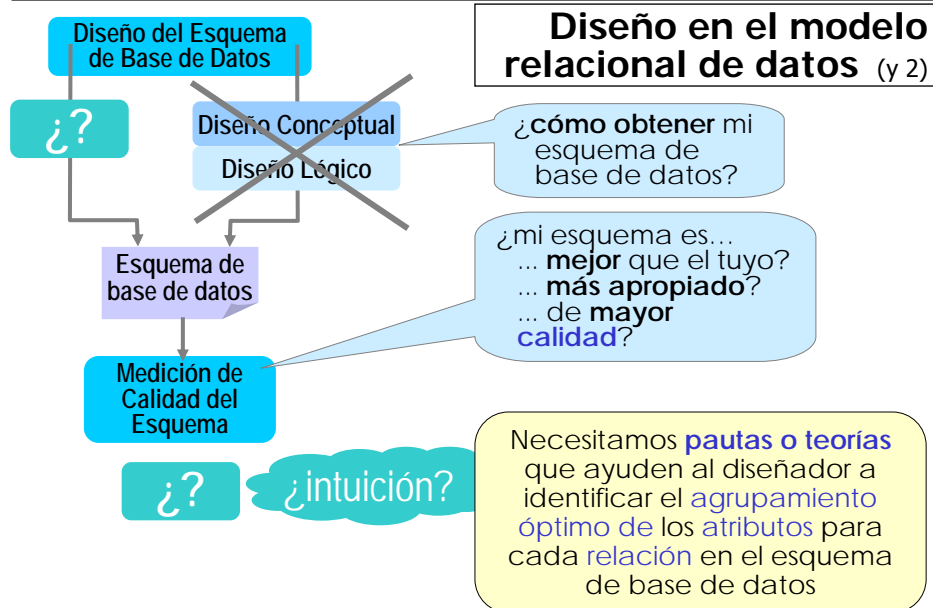
- Objetivo principal del desarrollo de un esquema de base de datos relacional:

Crear una representación precisa y adecuada de los datos, sus relaciones y sus restricciones

- Definición de **normalización** (E.F. Codd, 1972):

Técnica para producir un conjunto de relaciones (tablas) con propiedades deseables, a partir de los requisitos de datos de una organización

8.1 Motivación



8.1 Motivación

Dificultades en el diseño de esquemas relacionales

- **Propiedades no deseables** que puede tener un esquema de relación diseñado de forma incorrecta
 - **Repetición** de información
 - **Anomalías de actualización** de la base de datos
 - De **inserción**
 - De **modificación**
 - De **borrado**
 - **Incapacidad para representar** cierta información
 - **Pérdida de información** (se verá más adelante)
 - Aparición de **tuplas falsas**

8.1 Motivación

Dificultades en el diseño de esquemas relacionales (2)

- **Ejemplo 1:** información sobre **pedidos** (y productos y clientes) en una única relación

PEDIDO

nPed	codProd	descProd	precProd	uds	cifCli	nomCli	ciudCli	distanc
12	P1	silla	100	12	A12345678	Forte	Madrid	400
09	P1	silla	100	30	V22332233	Calero	Valencia	200
22	P1	silla	100	15	N43344334	Carazo	Alicante	80
15	P2	mesa	250	35	A12345678	Forte	Madrid	400
24	P2	mesa	250	20	V22332233	Calero	Valencia	200
05	P2	mesa	250	20	K11112222	Mitra	Madrid	400
11	P3	sillón	175	10	N43344334	Carazo	Alicante	80

...

8.1 Motivación

Dificultades en el diseño de esquemas relacionales (3)

- **Ejemplo 1** (continuación)

PEDIDO (nPed, codProd, descProd, precProd, uds, cifCli, nomCli, ciudCli, distanc)

- ¿Existe repetición de información?
- ¿Hay anomalías de actualización?
 - ¿Qué ocurre si se añade otro pedido, el nº 27, hecho por el cliente "Soriano" (CIF G22224444) de "Yecla" (a 100 km.), de 5 unidades del producto P3 con precio 175€?
 - ¿Qué pasa si el cliente "Forte" de "Madrid" se traslada a "Valencia"?
¿Y si el producto P1 aumenta de precio?
 - ¿Qué ocurre si eliminamos el producto P2 de la relación PEDIDO?
- ¿Existe incapacidad para representar información?
 - ¿Cómo representar (almacenar) un cliente que no ha realizado ningún pedido?
 - ¿Y un producto que todavía no ha sido pedido por ningún cliente?

8.1 Motivación

Dificultades en el diseño de esquemas relacionales (4)

- **Ejemplo 2:** información sobre **préstamos** (y sucursales) almacenada en una única relación

PRESTAMO

idSuc	ciudSuc	activo	nomCli	numPrest	importe
Centro	Arganzuela	10.818.215	Santos	P-17	1.200
Moralzarzal	La Granja	2.524.250	Gómez	P-23	2.400
Navacerrada	Aluche	2.043.440	López	P-15	1.800
Centro	Arganzuela	10.818.215	Soto	P-14	1.800
Becerril	Aluche	480.810	Santos	P-93	600
Collado	Aluche	9.616.194	Abril	P-11	1.080
Navas	Alcalá de H.	360.607	Valdivieso	P-29	1.440
Segovia	Cerceda	4.447.490	López	P-16	1.570
Centro	Arganzuela	10.818.215	González	P-63	2.400
Navacerrada	Aluche	2.043.440	Rodríguez	P-25	3.000
Galapagar	Arganzuela	8.534.375	Amol	P-10	2.640

8.1 Motivación

Dificultades en el diseño de esquemas relacionales (y 5)

- **Ejemplo 2** (continuación)

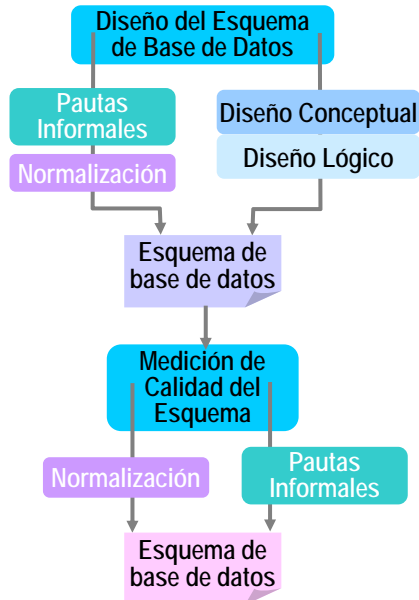
PRESTAMO (idSuc, ciudSuc, activo, nomCli, numPrest, importe)

- ¿Existe repetición de información?
- ¿Hay anomalías de actualización?
 - ¿Qué pasa si añadimos otro préstamo, P-31, por 1.803€, para el cliente Rodríguez, hecho por la sucursal "Navacerrada" (ubicada en "Aluche" y con activo de 2.043.440€)?
 - ¿Qué ocurre si la sucursal "Centro" se traslada a otra ciudad?
 - ¿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?

8.1 Motivación

Obtención de esquemas relacionales de calidad

- Para **crear buenos esquemas de BD** o para **medir su calidad...**
 - Seguir **pautas/criterios informales** de diseño
 - Emplear una técnica formal: **normalización**
- Se minimizará la aparición de las propiedades no deseables



8.1 Motivación

Obtención de esquemas relacionales de calidad (2)

Pautas de diseño de esquemas de relación

- **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 y tipos de relación (MERE) en una relación. Si una relación proviene de un único tipo de entidad o tipo de relación 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, hay que indicarlo 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).

8.1 Motivación

Obtención de esquemas relacionales de calidad (3)

Pautas de diseño de esquemas de relación

- Reducción de **nullos** 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, AVG...)
 - 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**
 - Esta pauta puede expresada de manera más formal (Conservación de la información -- 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 la aparición de tuplas falsas (espurias).

8.1 Motivación

Obtención de esquemas relacionales de calidad (y 4)

Normalización de esquemas de relación

- Proceso de **normalización**
 - Método **formal**, que identifica relaciones con base en su **clave primaria** (o candidatas en el caso de la FNBC) y las **dependencias funcionales** entre sus atributos
 - Emplea una serie de **pruebas**, descritas como **formas normales**, para identificar el agrupamiento óptimo de los atributos

8.2 Dependencias funcionales

Definición formal

- Sea R un esquema de relación, cuyo conjunto de atributos es A, y sean los descriptores (conjuntos de atributos) α y β , donde $\alpha \subseteq A$ y $\beta \subseteq A$, se dice que

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

Es decir, β **depende funcionalmente de** α si para cualesquiera dos tuplas t_1 y t_2 (de la extensión de una relación r con esquema R) que tienen los mismos valores en los atributos que forman α , ambas tuplas también tienen los mismos valores para los atributos que forman β

- También se dice que α determina funcionalmente a β
- Cada valor de α está asociado con exactamente un valor de β
- $\alpha \rightarrow \beta$ **no** significa que dado α pueda deducirse el valor de β
- Si $\alpha \rightarrow \beta$, **no sabemos nada** acerca de si $\beta \rightarrow \alpha$ o no

8.2 Dependencias funcionales

Ejemplo: esquema de relación EMP_DEP

● EMP_DEP (nomE, nssE, fechNacE, direcE, codD, nomD, nssJefeD)

Dependencias funcionales:

$$DF = \{ \begin{array}{l} nssE \rightarrow nomE, \\ nssE \rightarrow fechNacE, \\ nssE \rightarrow direcE, \\ codD \rightarrow nomD, \\ codD \rightarrow nssJefeD, \\ nssE \rightarrow codD, \\ nssJefeD \rightarrow codD \end{array} \}$$

8.2 Dependencias funcionales

Ejemplo: una posible extensión de la relación

nomE	nssE	fechNacE	direcE	codD	nomD	nssJefeD
Pérez	1	1979	Murcia	D1	Comercial	10
Gómez	4	1978	Madrid	D1	Comercial	10
Martínez	2	1980	Barcelona	D3	Producción	7
García	3	1980	Murcia	D2	I+D	5
González	5	1979	Murcia	D2	I+D	5
...

¿Se podría incluir la siguiente tupla? ¿Por qué?

Pérez	1	1979	Murcia	D2	I+D	5
-------	---	------	--------	----	-----	---

8.2 Dependencias funcionales

- En general, una **dependencia funcional** (df) $\alpha \rightarrow \beta \dots$
 - Es una **restricción entre 2 conjuntos de atributos** α y β
 - Es una **propiedad de la semántica de los atributos** de un esquema de relación R:
 - ◀ Definida por alguien que conoce bien dicha semántica
 - Es una **propiedad del esquema** de relación R, no del contenido (estado, extensión)
 - **Se debe cumplir** para cualquier extensión de una relación r con esquema de relación R, $r(R)$
 - Al **observar una extensión** concreta de una relación $r(R) \dots$
 - No se puede **demostrar** una df, pero sí se puede afirmar
 - Sólo se puede **deducir que no existe** (no se cumple) una df
 - **Extensiones permitidas** (válidas) y **no permitidas**
 - Si cierta df es cierta para un esquema de relación R, entonces **cualquier extensión de una relación $r(R)$ que no cumple dicha df, será una extensión no permitida** (no válida)

8.2 Dependencias funcionales

- **Determinante**

- Atributo o conjunto de **atributos** en el **lado izquierdo** de la flecha que describe una dependencia funcional

- **Dependencia funcional completa**

- $\alpha \rightarrow \beta$ es completa si la **parte derecha** β **no depende de ningún subconjunto del determinante** α
 - ▶ β depende de α en su conjunto

- **Dependencia funcional transitiva**

- Sean α , β y γ tres descriptores de una relación, la df $\beta \rightarrow \gamma$ es transitiva si el conjunto de dependencias funcionales de la relación también incluye las df $\alpha \rightarrow \beta$ y $\alpha \rightarrow \gamma$
- Se dice entonces que γ depende transitivamente de α a través de β (supuesto que α no sea dependiente funcionalmente de β o γ)

codprofesor \rightarrow coddepartamento

codprofesor \rightarrow codfacultad

coddepartamento \rightarrow codfacultad ◀

8.2 Dependencias funcionales

Reglas de inferencia

- Dado un **conjunto de dependencias funcionales** DF que R satisface, podemos **deducir que R cumple otras df**

- Serán df implicadas lógicamente por DF
- No es necesario conocer la extensión de ninguna relación $r(R)$

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, en DF sólo se especifican las **df obvias** para un esquema de relación R: las **determinadas** sin dificultad **a partir de la semántica de los atributos de R**

- A partir de ellas, se puede inferir el conjunto completo de df que se cumplen en R
- Esto es esencial para el diseño de buenos esquemas de relación

8.2 Dependencias funcionales

Axiomas de Armstrong

Sean los 4 descriptores $\alpha, \beta, \gamma, \delta$, todos subconjuntos de A

- **Regla de reflexividad**

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

dependencia
funcional trivial

- **Regla de aditividad** (de 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

8.2 Dependencias funcionales

Otras reglas de inferencia

... deducibles a partir de los Axiomas de Armstrong

- **Regla de unión** (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 $\alpha \rightarrow \gamma$

👁 sólo para la
parte derecha
de una df!!

- **Regla de pseudotransitividad**

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

- **Regla de composición**

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

8.2 Dependencias funcionales

Cierre de un conjunto de dependencias funcionales

- El **cierre** de DF (DF^+) es el conjunto de todas las df que puedan ser inferidas a partir de DF, aplicando los axiomas de Armstrong
- $\alpha \rightarrow \beta$ se infiere de un conjunto de df DF de R si toda extensión de $r(R)$ 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...

8.2 Dependencias funcionales

Cierre de un conjunto de dependencias funcionales (2)

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

1. Establecer los conjuntos de atributos α que son los determinantes de las df de DF
2. Para cada α , aplicar los Axiomas de Armstrong para determinar qué atributos dependen funcionalmente de α
 - Esto es el cálculo del **cierre de un descriptor** α sobre un conjunto de dependencias funcionales DF, denotado por α^+_{DF}
3. Formar DF^+ mediante la unión de todas las dependencias funcionales descritas por los cierres de descriptors obtenidos en el paso 2

8.2 Dependencias funcionales

Cierre de un conjunto de dependencias funcionales (3)

Cálculo del **cierre de un descriptor** α^+_{DF}

$$\alpha^+ = \alpha$$

REPETIR HASTA QUE α^+ NO CAMBIE

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

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

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

- Sea $R_1(A_1, DF_1)$, donde $A_1=\{a,b,c,g,h,i\}$ y $DF_1=\{a \rightarrow b, a \rightarrow c, cg \rightarrow h, cg \rightarrow i, b \rightarrow h\}$
Calcule $\{ag\}^+_{DF1}$ y DF_1^+

- Sea $R_2(A_2, DF_2)$, donde $A_2=\{a,b,c,d,e,f\}$ y $DF_2=\{ab \rightarrow c, bc \rightarrow ad, d \rightarrow e, cf \rightarrow b\}$
Calcule $\{ab\}^+_{DF2}$ y DF_2^+

8.2 Dependencias funcionales

Cierre de un conjunto de dependencias funcionales (y 4)

Comprobación del **cumplimiento de una df** en DF

- La dependencia funcional $\alpha \rightarrow \beta$ **se cumple en** un conjunto DF de dependencias funcionales **si** $\alpha \rightarrow \beta \in DF^+$

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

– Comprobar que $\alpha \rightarrow \beta$ se cumple en DF es ver **si β depende funcionalmente de α con base en las df de DF**

– Como α^+_{DF} contiene todo atributo que depende funcionalmente de α con base en DF, sólo habrá que comprobar si β está entre ellos

- 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:

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

8.2 Dependencias funcionales

Determinación de las claves de un esquema de relación

- Hasta ahora hemos estudiado el **concepto intuitivo de (super)clave**
 - Descriptor que **identifica unívocamente cada tupla en una relación**
- Definición **formal** de clave
 - Un descriptor α es clave **si todos los atributos de la relación dependen funcionalmente de él**
 - Sea $R(A, DF)$, el descriptor $\alpha \subseteq A$ es clave si $(\alpha)^+_{DF} = A$
 - Es decir, su cierre es el conjunto completo de atributos de R
 - Esto asegura que no hay dos tuplas distintas con igual valor para α

8.2 Dependencias funcionales

Determinación de claves de un esquema de relación

- **Determinación del conjunto K de (super)claves** de $R(A, DF)$
 - $K = A$
 - PARA CADA $\alpha \in K$ (todos los subconjuntos posibles de A)
 - SI $(K - \alpha)^+_{DF} = A$, ENTONCES $K = K - \alpha$
 - Tras esto, se puede aplicar el algoritmo siguiente (a partir del paso 3), para ver cuáles de las claves encontradas son claves candidatas (mínimas) en R

8.2 Dependencias funcionales

Determinación de si un descriptor es clave

- **Determinación de si α es clave candidata** de $R(A, DF)$
 1. CALCULAR α^+_{DF}
 2. Si $\alpha^+_{DF} = A$, ENTONCES α es superclave,
SI NO, α no es clave
 3. Si α es superclave, ENTONCES
calcular todos los subconjuntos α' de α
REPETIR PARA CADA α'
SI $\alpha'^+_{DF} = A$, ENTONCES α no es clave candidata
SI ninguna α' cumple lo anterior ENTONCES α es clave candidata

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

8.2 Dependencias funcionales

Recubrimiento canónico o minimal

- Conjunto de dependencias funcionales **simplificado, DF^m**
 - Toda df tiene **un solo atributo en la parte derecha**
 - **No tiene atributos extraños**: no se puede quitar un atributo de una df en DF^m y obtener un conjunto de df equivalente a DF^m
 - **No tiene dependencias funcionales redundantes**: no se puede quitar una df de DF^m y obtener un conjunto de df equivalente a DF^m
- El recubrimiento canónico o minimal de un conjunto de dependencias funcionales DF es equivalente al DF original

8.2 Dependencias funcionales

Recubrimiento canónico o minimal

- **Atributo extraño** en una dependencia funcional de DF
 - Sea una df $\alpha \rightarrow \beta$ en DF,
 - $a \in \alpha$ es un **atributo extraño** si $(\alpha - a) \rightarrow \beta \in DF^+$
 - ▶ El atributo a puede ser eliminado sin modificar el cierre de DF
- **Dependencia funcional redundante** de DF
 - $\alpha \rightarrow \beta$ es **redundante** si puede derivarse de $G = \{DF - (\alpha \rightarrow \beta)\}$ es decir, si $\beta \in \alpha^+_G$
 - ▶ Una df redundante puede ser eliminada sin modificar el cierre de DF

8.2 Dependencias funcionales

Recubrimiento canónico o minimal (2)

- Es interesante utilizar el recubrimiento minimal de un conjunto de dependencias funcionales porque...
 - Toda **dependencia funcional es una restricción de integridad** (semántica). Si no hay df redundantes, se **minimiza el coste de mantenimiento de la integridad** de la BD sin disminuir la semántica
 - DF^m **se utiliza para normalizar** una relación y para **calcular las claves**, así que se **disminuye el coste de los algoritmos** empleados para ello

8.2 Dependencias funcionales

Recubrimiento canónico o minimal (3)

0. Transformar toda df del conjunto DF de forma que...
 - Sólo tenga un atributo en el implicado (parte derecha)
 - Sea no trivial
1. Eliminar atributos extraños
REPETIR PARA CADA $\alpha \rightarrow b$ EN DF
 $L = \alpha$
 REPETIR POR CADA ATRIBUTO a DE α
 SI $b \in (\alpha - a)^+_{DF}$ ENTONCES $L = L - a$
 REEMPLAZAR $\alpha \rightarrow b$ POR $L \rightarrow b$ EN DF
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. $DF^m = H$

8.2 Dependencias funcionales

Recubrimiento canónico o minimal (y 4)

- 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
- Sea $LIBRO(codlibro, isbn, editorial, pais)$
 donde $DF=\{ codlibro \rightarrow isbn editorial,$
 $isbn \rightarrow codlibro editorial pais,$
 $editorial \rightarrow pais\}$
 Calcule DF^m
- El recubrimiento minimal DF^m no tiene por qué ser único
- Sea $R(a, b, c)$
 donde $DF=\{ a \rightarrow bc,$
 $b \rightarrow ac,$
 $c \rightarrow ab\}$
 Calcule los tres DF^m que se pueden obtener

8.3 Formas normales

- La normalización es una técnica formal para analizar relaciones con base en sus claves candidatas y dependencias funcionales, que aplica una serie de pruebas (descritas como **formas normales**) a las relaciones individuales, con el fin de identificar un conjunto de relaciones que soporten adecuadamente los requisitos de datos de la organización
- El esquema de la base de datos puede ser **normalizado hasta una** determinada **forma normal** deseada
- **Formas normales básicas**
 - 1FN
 - 2FN, 3FN
 - FNBC (R. Boyce y E.F. Codd, 1974)
- **Otras formas normales**
 - 4FN, 5FN (Fagin, 1977, 1979)

► Basadas en dependencias funcionales entre atributos de un esquema de relación

8.3 Formas normales

- Sea el esquema de relación $R(A, DF)$ **mal diseñado**
 - Incumple alguna forma normal
- será necesario **descomponer** R en un **conjunto de n esquemas** de relación
- $$\{ R_i (A_i, DF_i) \} i:1..n$$
- que cumpla** las propiedades de
- **Mínima redundancia** de datos
 - **Conservación de la información**
 - **Conservación de las dependencias**

8.3 Formas normales

Minimización de la redundancia

- Los esquemas R_i resultado de la descomposición de R , deben estar **en una forma normal superior a la de R**

$$\textcircled{1} 1\text{FN} < 2\text{FN} < 3\text{FN} < \text{FNBC} < 4\text{FN} < 5\text{FN}$$

- ▶ Así se reduce al mínimo la redundancia
 - Separando atributos en relaciones distintas
- ▶ Y, por tanto, se evitan propiedades no deseables
 - Desperdicio de espacio (repetición)
 - Anomalías de actualización
 - Incapacidad para representar cierta información
 - ...

8.3 Formas normales

Minimización de la redundancia: 1ª Forma Normal

- R está en 1FN si **ningún atributo es multivalorado** (puede tener más de un valor a la vez) **ni tiene un valor compuesto** (dominio no atómico)

- Es una propiedad inherente al modelo relacional de datos

LIBRO (esquema **incorrecto**: no está en 1FN)

codigo	titulo	autor
9090	Fundamentos de sistemas de bases de datos	Elmasri , Navathe
8070	Diseño de bases de datos relacionales	de Miguel, Piattini, Marcos

Multivalorado

LIBRO (esquema en 1FN)

Redundancia

codigo	titulo	autor
9090	Fundamentos de sistemas de bases de datos	Elmasri
9090	Fundamentos de sistemas de bases de datos	Navathe
8070	Diseño de bases de datos relacionales	de Miguel
8070	Diseño de bases de datos relacionales	Piattini
8070	Diseño de bases de datos relacionales	Marcos

8.3 Formas normales

Minimización de la redundancia: 2ª Forma Normal

- ① Un **atributo principal** es un 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 cualquier clave candidata**
 - Es decir, **no existe un atributo no principal que dependa funcionalmente de parte de una clave**

📁 PUBLICA (artículo, revista, numero, pagina, editorial)

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

PUBLICA no está en 2FN, ¿por qué?

- Un artículo puede aparecer en varias revistas diferentes
- En cada revista, un artículo puede aparecer en una y sólo una página
- Una revista publica varios artículos
- Una revista es editada por una editorial

8.3 Formas normales

Minimización de la redundancia: 3ª Forma Normal

- Varias **definiciones** equivalentes entre sí:
- 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 ninguna clave candidata

8.3 Formas normales

Minimización de la redundancia: 3ª Forma Normal (y 2)

□ 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, $\beta \subseteq \alpha$
- α es una **superclave** del esquema R, es decir $\alpha^+_{DF} = A$
- **Cada atributo a en $\beta - \alpha$ es principal** (está contenido en alguna clave candidata de R)

- SOCIO (dni, ciudad, país) $DF = \{ \text{dni} \rightarrow \text{ciudad}, \text{dni} \rightarrow \text{país}, \text{ciudad} \rightarrow \text{país} \}$
La relación SOCIO no está en 3FN, ¿por qué?

8.3 Formas normales

Minimización de redundancia: FN de Boyce-Codd

□ La Forma Normal de Boyce-Codd, **FNBC**, es una definición más estricta de la 3FN

- $R(A, DF)$ 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, $\beta \subseteq \alpha$
 - α es una **superclave** del esquema R, es decir $\alpha^+_{DF} = A$
- Es decir, R está en FNBC si **todo determinante es una clave candidata**

8.3 Formas normales

Minimización de redundancia: FN de Boyce-Codd (2)

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

8.3 Formas normales

Minimización de redundancia: FN de Boyce-Codd (y 3)

● ARTICULO1 (codart, titulo, revista, numero, pagina)

DF = { codart → titulo,

Un título nunca se repite en una misma revista

titulo → codart,

Un artículo puede publicarse en varias revistas

codart revista numero → pagina;

titulo revista numero → pagina;

pagina revista numero → codart, titulo }

Está en 3FN, pero **no** en FNBC

Claves de ARTICULO1:

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

Claves de ARTICULO2:

CK1 = { revista, numero, codart }

CK2 = { revista, numero, pagina }

8.3 Formas normales

Minimización de redundancia

- Las siguientes afirmaciones son ciertas
 - Todo esquema **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 2 atributos** **está siempre en FNBC**
 - Todo esquema **R con 3 atributos y una sola clave compuesta por 2 atributos**, **está en FNBC**
- Intente comprobar la veracidad de estas afirmaciones...

8.4 Descomposición de relaciones

Conservación de la información

- Ya vimos que la relación PRESTAMO adolecía de propiedades no deseables

```
PRESTAMO ( idSuc, ciudSuc, activo, nomCli, numPrest, importe )
```

donde $DF = \{ idSuc \rightarrow activo\ ciudSuc, \\ numPrest \rightarrow importe\ idSuc\ nomCli \}$
- La solución es **descomponer la relación en dos esquemas**
 - Dos **proyecciones** de PRESTAMO
- Pero **no sirve cualquier descomposición...**
... lo veremos con un ejemplo:

```
SUCURSAL_CLIENTE ( idSuc, ciudSuc, activo, nomCli )
```

```
PRESTAMO_CLIENTE ( nomCli, numPrest, importe )
```

8.4 Descomposición de relaciones

Conservación de la información

Contenido original de la relación PRESTAMO

idSuc	ciudSuc	activo	nomCli	numPrest	importe
Centro	Arganzuela	10.818.215	Santos	P-17	1.200
Moralzarzal	La Granja	2.524.250	Gómez	P-23	2.400
Navacerrada	Aluche	2.043.440	López	P-15	1.800
Centro	Arganzuela	10.818.215	Soto	P-14	1.800
Becerril	Aluche	480.810	Santos	P-93	600
Collado	Aluche	9.616.194	Abril	P-11	1.080
Navas	Alcalá de H.	360.607	Valdivieso	P-29	1.440
Segovia	Cerceda	4.447.490	López	P-16	1.570
Centro	Arganzuela	10.818.215	González	P-63	2.400
Navacerrada	Aluche	2.043.440	Rodríguez	P-25	3.000
Galapagar	Arganzuela	8.534.375	Amol	P-10	2.640

8.4 Descomposición de relaciones

Conservación de la información

SUCURSAL_CLIENTE

idSuc	ciudSuc	activo	nomCli
Centro	Arganzuela	10.818.215	Santos
Moralzarzal	La Granja	2.524.250	Gómez
Navacerrada	Aluche	2.043.440	López
Centro	Arganzuela	10.818.215	Soto
Becerril	Aluche	480.810	Santos
Collado	Aluche	9.616.194	Abril
Navas	Alcalá de H.	360.607	Valdivieso
Segovia	Cerceda	4.447.490	López
Centro	Arganzuela	10.818.215	González
Navacerrada	Aluche	2.043.440	Rodríguez
Galapagar	Arganzuela	8.534.375	Amol

8.4 Descomposición de relaciones

Conservación de la información

PRESTAMO_CLIENTE

<u>nomCli</u>	<u>numPrest</u>	<u>importe</u>
Santos	P-17	1.200
Gómez	P-23	2.400
López	P-15	1.800
Soto	P-14	1.800
Santos	P-93	600
Abril	P-11	1.080
Valdivieso	P-29	1.440
López	P-16	1.570
González	P-63	2.400
Rodríguez	P-25	3.000
Amol	P-10	2.640

8.4 Descomposición de relaciones

Conservación de la información

Consulta: obtener las sucursales con préstamos con importe < 1.200€

- Es necesario **reconstruir** la relación PRESTAMO, realizando un JOIN **entre SUCURSAL_CLIENTE y PRESTAMO_CLIENTE, vía el atributo nomCli**

```
SELECT idSuc
FROM Sucursal_Cliente SC, Préstamo_Cliente PC
WHERE SC.nomCli=PC.nomCli AND PC.importe < 1200;
```

Esta sentencia es equivalente a esta otra:

```
SELECT idSuc
FROM Sucursal_Cliente NATURAL JOIN Préstamo_Cliente
WHERE importe < 1200;
```

- **Resultado obtenido:** tres tuplas ((Centro...), (Becerril...), (Collado...))
- ¡ Pero el **resultado correcto** es: ((Becerril...), (Collado...)) !



8.4 Descomposición de relaciones

Conservación de la información

SUCURSAL_CLIENTE NATURAL JOIN PRESTAMO_CLIENTE

idSuc	ciudSuc	activo	nomCli	numPrest	importe
Centro	Arganzuela	10.818.215	Santos	P-17	1.200
Centro	Arganzuela	10.818.215	Santos	P-93	600
Moralzarzal	La Granja	2.524.250	Gómez	P-23	2.400
Navacerrada	Aluche	2.043.440	López	P-15	1.800
Navacerrada	Aluche	2.043.440	López	P-16	1.570
Centro	Arganzuela	10.818.215	Soto	P-14	1.800
Becerril	Aluche	480.810	Santos	P-17	1.200
Becerril	Aluche	480.810	Santos	P-93	600
Collado	Aluche	9.616.194	Abril	P-11	1.080
Navas	Alcalá de H.	360.607	Valdivieso	P-29	1.440
Segovia	Cerceda	4.447.490	López	P-15	1.800
Segovia	Cerceda	4.447.490	López	P-16	1.570
Centro	Arganzuela	10.818.215	González	P-63	2.400
Navacerrada	Aluche	2.043.440	Rodríguez	P-25	3.000
Galapagar	Arganzuela	8.534.375	Amol	P-10	2.640

Tuplas Falsas
(espurias)

Tema 8. Teoría de la normalización

51

8.4 Descomposición de relaciones

Conservación de la información

- Hay **más tuplas** en el resultado de la operación de reunión SUCURSAL_CLIENTE NATURAL JOIN PRESTAMO_CLIENTE que en la relación original PRESTAMO, pero existe **menos información**:

descomposición de reunión con pérdida

- ☹ Una **descomposición descuidada** puede conducir a otro **diseño incorrecto**, debido a la **pérdida de información**

- ¿Cuál es la causa de esta pérdida de información?
- La siguiente descomposición de PRESTAMO es de reunión **sin** pérdida
¿Por qué?

SUCURSAL (idSuc, ciudSuc, activo)

PRESTAMO (idSuc, nomCli, numPrest, importe)

Tema 8. Teoría de la normalización

52

8.4 Descomposición de relaciones

Conservación de la información

- La información contenida en $R(A, DF)$ debe ser la misma que la contenida en las $R_i(A_i, DF_i)$; para ello, debe conseguirse...
 - **Conservación de los atributos** $\cup A_i = A$
 - **Conservación del contenido** (tuplas o extensión) $*r_i = r$
 - $\forall r$ extensión de R , la **reunión natural de extensiones** r_i de R_i , **debe producir la r origen**: reconstrucción de la extensión original
 - Si no se conserva el contenido, aparecen **tuplas falsas**
- Una descomposición de **reunión con pérdida puede no ser detectada** para algunas extensiones de las R_i
 - Un JOIN entre ciertas extensiones puede no generar tuplas falsas
- ▶ Es necesario encontrar una forma de **descomponer que siempre conserve la información**
 - **Descomposición en proyecciones independientes**

8.4 Descomposición de relaciones

Conservación de la información

Descomposición 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**
 - Sus **atributos comunes** son la **clave primaria** de, al menos, una de las dos proyecciones
 - Cada **dependencia funcional** en R puede deducirse de las dependencias funcionales de R_1 y R_2
- **PELICULA**(titulo,año,duracion,tipo,estudio,actor), $DF=\{\text{titulo año} \rightarrow \text{duracion tipo estudio}\}$
 - Redundancia: se repite todo por cada actor participante en la película
 - Causa: transformación de un atributo multivalorado (actor)
 - No está en FNBC porque (titulo, año) no es clave; la clave es (titulo, año, actor)
 - ▶ **Descomposición en proyecciones independientes**:
 - $R_1(\text{titulo, año, duracion, tipo, estudio})$ con $DF_1=\{\text{titulo año} \rightarrow \text{duracion tipo estudio}\}$
 - $R_2(\text{titulo, año, actor})$, con DF_2 vacío

8.4 Descomposición de relaciones

Conservación de la información

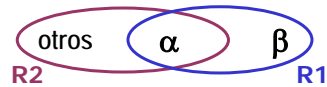
Descomposición en proyecciones independientes (2)

- Algoritmo de descomposición

Tomar $\alpha \rightarrow \beta$ de DF^+ no trivial, que viole la FNBC (o 3FN)

Obtener dos proyecciones de $R(A, DF)$

$R1(\alpha, \beta, DF1)$ y $R2(A - \beta, DF2)$,



donde $DF1$ y $DF2$ son conjuntos de dfs sobre $R1$ y $R2$ respectivamente

Si $R2$ (o $R1$) no está en FNBC (o 3FN), descomponerla de nuevo

Guía en el proceso de descomposición
en proyecciones independientes:
Utilizar las dependencias funcionales
que incumplan la FN deseada

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

8.4 Descomposición de relaciones

Conservación de la información

Descomposición en proyecciones independientes (y 3)

- Las dos proyecciones independientes resultantes **cumplen** uno de estos **principios de descomposición sin pérdidas**:

la $df((R_1 \cap R_2) \rightarrow (R_1 - R_2)) \in DF^+$ o bien

la $df((R_1 \cap R_2) \rightarrow (R_2 - R_1)) \in DF^+$

- Siempre es posible **descomponer** una relación en dos **proyecciones independientes** para llegar a la **3FN**

- Esto **no siempre es posible** para llegar a la **FNBC**, pues a veces se pierden dependencias funcionales

• PROYECTA($\text{pelicula}, \text{cine}, \text{ciudad}$), $DF = \{\text{pelicula ciudad} \rightarrow \text{cine}, \text{cine} \rightarrow \text{ciudad}\}$

• NOTA($\text{dniProf}, \text{nomProf}, \text{dniAlu}, \text{calific}$)

$DF = \{\text{dniProf} \rightarrow \text{nomProf}, \text{nomProf} \rightarrow \text{dniProf}, \text{dniProf dniAlu} \rightarrow \text{calific}\}$

8.4 Descomposición de relaciones

Conservación de las dependencias funcionales

- La descomposición de R en los R_i también debe conservar el conjunto de dependencias funcionales DF de R, pues cada dependencia funcional es una **restricción de integridad** que refleja semántica del mundo real
$$(\cup DF_i)^+ = DF^+$$
- El conjunto de dependencias de origen es **equivalente** a la unión de los conjuntos de dependencias de los esquemas resultantes
- De este modo se consigue que cualquier actualización de la base de datos obtenga siempre una relación legal (cuya extensión tras la actualización cumple todas las dependencias funcionales dadas)
- La **descomposición en proyecciones independientes garantiza la conservación de las dependencias funcionales**

8.4 Descomposición de relaciones

Conservación de las dependencias funcionales

- **Determinación de las dependencias funcionales que cumple cada proyección** resultante de una descomposición
Sea $R(A, DF)$ y sea $R_1(A_1, DF_1)$ una de las proyecciones de R, ¿DF1?
 - Considerar **cada descriptor α** , subconjunto de A_1
 - **Calcular α^+** , respecto del conjunto de dependencias DF (#)
 - Para **cada atributo** a tal que
 - a es un atributo de R_1 ($a \in A_1$)
 - a está contenido en α^+ ($a \in \alpha^+$)
 - a no está contenido en α ($a \notin \alpha$)la df $\alpha \rightarrow a$ se cumple en R_1 , es decir $(\alpha \rightarrow a) \in DF_1$
- (#) No es necesario determinar el cierre de ...
- todo el conjunto de atributos de R_1 (es decir, si $\alpha = A_1$)
 - los α que no contengan la parte izquierda de alguna df
 - los α que contengan un atributo que no sea parte izquierda de alguna df

8.4 Descomposición de relaciones

Un ejemplo

- Otra descomposición adecuada de PRESTAMO es la siguiente:
SUCURSAL(idSuc, activo, ciudSuc) DF={ idSuc → activo ciudSuc }
PRESTAMO(idSuc, numPrest, importe) DF={ numPrest → importe idSuc }
CLIENTE(nomCli, numPrest) DF={ numPrest → nomCli, ... } 👁
 - Esta descomposición es la más adecuada, si consideramos que para cada cliente se necesitará almacenar más atributos, como nif, direccion, etc., que en el caso de permanecer en PRESTAMO, implicarían propiedades no deseables
- Esta descomposición cumple las propiedades de **conservación de la información**, **conservación de las dependencias funcionales** y se **minimiza la redundancia** de datos
- Por tanto, los nuevos esquemas son **equivalentes** a, y **mejores** que, el esquema de relación original PRESTAMO

8.5 Enfoques de diseño relacional

- Dos enfoques de diseño: **análisis** y **síntesis**
- La teoría de la normalización puede emplearse para...
 - **Diseñar un esquema de base de datos relacional:**
Se parte de una única relación (con toda la información) y sus dependencias funcionales, y...
 - se aplica **síntesis**, o bien
 - se aplica **análisis**... hasta llegar a un conjunto de relaciones correcto (en 3FN o FNBC).
 - **Evaluar la calidad de un esquema** obtenido directamente o por transformación de un esquema conceptual (MERE):
Se comprueba en qué FN está cada esquema de relación
Si el diseño no es adecuado (algunas relaciones no están en 3FN o FNBC), se puede aplicar **análisis** sobre las relaciones mal diseñadas para obtener las relaciones adecuadas

8.5 Enfoques de diseño relacional

Análisis o descomposición: diseño descendente

- Se parte de un **esquema de relación** R y de su conjunto de dependencias funcionales, DF
 - Se **descompone R** sucesivamente **en proyecciones que conservan**
 - la **información**
 - las **dependencias**
 - Los **esquemas resultantes** cada vez ...
 - **tienen menos atributos** (menor grado)
 - están en **formas normales más avanzadas**
 - **Fin** del proceso ...
 - Al llegar a la **FN deseada**, o
 - Si **seguir implica** la **pérdida de dependencias**
- Separa información referente a conceptos distintos
- Empleo de árboles de descomposición

8.5 Enfoques de diseño relacional: Análisis

Proceso de análisis o descomposición de R(A,DF)

1. Determinar el **recubrimiento minimal** DF^m
2. Determinar las **claves** de R
3. Determinar el **nivel de normalización** de R
4. Si R no está en la FN deseada (3FN o FNBC),
obtener **proyecciones independientes** y parar cuando...
se alcance la **FN deseada** o
seguir descomponiendo **supusiera pérdidas**

8.5 Enfoques de diseño relacional: Análisis

Ejemplo 1

PRESTAMO(codSocio, nomSocio, codLibro, fechPrest, editorial, pais)

$DF^m = \{ \text{codSocio} \rightarrow \text{nomSocio},$
 $\text{nomSocio} \rightarrow \text{codSocio},$
 $\text{codLibro} \rightarrow \text{editorial},$
 $\text{editorial} \rightarrow \text{pais},$
 $\text{codSocio codLibro} \rightarrow \text{fechPrest} \}$

Claves: $K = \{(\text{codSocio}, \text{codLibro}), (\text{nomSocio}, \text{codLibro})\}$

¿Cuál es su nivel de normalización?

Se desea obtener un conjunto de esquemas de relación en **FNBC**

8.5 Enfoques de diseño relacional: Análisis

Ejemplo 1: descomposición del esquema PRESTAMO

R1(editorial, pais) $K1 = (\text{editorial})$ $DF1 = \{ \text{editorial} \rightarrow \text{pais} \}$ ●

R2(codSocio, nomSocio, codLibro, fechPrest, editorial)

$K2 = \{(\text{codSocio}, \text{codLibro}), (\text{nomSocio}, \text{codLibro})\}$

$DF2 = \{ \text{codSocio} \rightarrow \text{nomSocio}, \text{nomSocio} \rightarrow \text{codSocio}, \text{codLibro} \rightarrow \text{editorial},$
 $\text{codSocio codLibro} \rightarrow \text{fechPrest} \}$

R1 está en FNBC, pero R2 no, así que **ha de ser descompuesta**:

R3(codLibro, editorial) $K3 = (\text{codLibro})$ $DF3 = \{ \text{codLibro} \rightarrow \text{editorial} \}$ ●

R4(codSocio, nomSocio, codLibro, fechPrest)

$K4 = \{(\text{codSocio}, \text{codLibro}), (\text{nomSocio}, \text{codLibro})\}$

$DF4 = \{ \text{codSocio} \rightarrow \text{nomSocio}, \text{nomSocio} \rightarrow \text{codSocio},$
 $\text{codSocio codLibro} \rightarrow \text{fechPrest} \}$

R3 está en FNBC, pero R4 no, así que **ha de ser descompuesta**:

R5(codSocio, nomSocio) $K5 = \{(\text{codSocio}), (\text{nomSocio})\}$ ●

$DF5 = \{ \text{codSocio} \rightarrow \text{nomSocio}, \text{nomSocio} \rightarrow \text{codSocio} \}$

R6(codSocio, codLibro, fechPrest) $K6 = \{(\text{codSocio}, \text{codLibro})\}$ ●

$DF6 = \{ \text{codSocio codLibro} \rightarrow \text{fechPrest} \}$

8.5 Enfoques de diseño relacional: Análisis

Ejemplo 1: resultado de la descomposición

●(relaciones R1, R3, R5 y R6 renombradas)

EDITORIAL (editorial, pais)

$K = (\text{editorial})$

$DF = \{ \text{editorial} \rightarrow \text{pais} \}$

LIBRO (codLibro, editorial)

$K = (\text{codLibro})$

$DF = \{ \text{codLibro} \rightarrow \text{editorial} \}$

SOCIO (codSocio, nomSocio)

$K = \{ (\text{codSocio}), (\text{nomSocio}) \}$

$DF = \{ \text{codSocio} \rightarrow \text{nomSocio}, \text{nomSocio} \rightarrow \text{codSocio} \}$

PRESTAMO (codSocio, codLibro, fechPrest)

$K = \{ (\text{codSocio}, \text{codLibro}) \}$

$DF = \{ \text{codSocio codLibro} \rightarrow \text{fechPrest} \}$

8.5 Enfoques de diseño relacional: Análisis

Ejemplo 2: pérdidas de dfs en el proceso de descomposición

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

$R1(b, a, c) K1 = (b) DF1 = \{ b \rightarrow a, c \}$

$R2(b, d, e, f, g) K2 = (b, e, g) DF2 = \{ e \rightarrow d, d \rightarrow f \}$

R1 está en FNBC, pero R2 no, así que ha de ser descompuesta:

$R3(e, d) K3 = (e) DF3 = \{ e \rightarrow d \}$

$R4(b, e, f, g) K4 = (b, e, g) DF4 = \emptyset$

👉 R2 debe ser descompuesta de otro modo: se ha perdido la df $d \rightarrow f$

$R3(d, f) K3 = (d) DF3 = \{ d \rightarrow f \}$

$R4(b, d, e, g) K4 = (b, e, g) DF4 = \{ e \rightarrow d \}$

R3 está en FNBC, pero R4 no, así que ha de ser descompuesta:

$R5(e, d) K5 = (e) DF5 = \{ e \rightarrow d \}$

$R6(b, e, g) K6 = (b, e, g) DF6 = \emptyset$

Conjunto de esquemas de relación resultado: R1, R3, R5, R6 (todos en FNBC)

8.5 Enfoques de diseño relacional: Análisis

Ejemplo 3

- Aplicar el algoritmo para dejar en **FNBC** el siguiente esquema de relación

CUENTA (idsuc, ciudsuc, activo, nomcli, codcuenta, saldo)

donde $DF^m = \{ \text{idsuc} \rightarrow \text{activo ciudsuc}, \text{codcuenta} \rightarrow \text{idsuc saldo} \}$

y $K = (\text{codcuenta}, \text{nomcli})$

$R1(\text{idsuc}, \text{activo}, \text{ciudsuc}); DF1 = \{ \text{idsuc} \rightarrow \text{activo ciudsuc} \}; K1 = (\text{idsuc})$

$R2(\text{idsuc}, \text{nomcli}, \text{codcuenta}, \text{saldo}); DF2 = \{ \text{codcuenta} \rightarrow \text{idsuc saldo} \};$
 $K2 = (\text{nomcli}, \text{codcuenta})$

$R1$ está en FNBC, pero $R2$ no, así que se sustituye por

$R3(\text{idsuc}, \text{codcuenta}, \text{saldo}); DF3 = \{ \text{codcuenta} \rightarrow \text{saldo idsuc} \}; K3 = (\text{codcuenta})$

$R4(\text{nomcli}, \text{codcuenta}); DF4 = \emptyset; K4 = (\text{nomcli}, \text{codcuenta})$

$R3$ y $R4$ están en FNBC

- El conjunto de esquemas de relación resultante es $\{ R1, R3, R4 \}$, todos en FNBC

8.5 Enfoques de diseño relacional: Análisis

Ejemplo 4

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

BANQUERO_PERSONAL(idsuc, nomcli, nombanq)

donde $DF^m = \{ \text{nombanq} \rightarrow \text{idsuc}, \text{idsuc nomcli} \rightarrow \text{nombanq} \}$

y las claves $K = \{ (\text{idsuc}, \text{nomcli}), (\text{nomcli}, \text{nombanq}) \}$

$R1(\text{nombanq}, \text{idsuc}); DF1 = \{ \text{nombanq} \rightarrow \text{idsuc} \}; K1 = (\text{nombanq})$

$R2(\text{nomcli}, \text{nombanq}); DF2 = \emptyset; K2 = (\text{nomcli}, \text{nombanq})$

$R1$ y $R2$ están en FNBC

- El conjunto de esquemas resultante es $\{ R1, R2 \}$, todos en FNBC, pero...
 - sólo preserva $\text{nombanq} \rightarrow \text{idsuc}$ (además de las df triviales) y
 - no conserva $\text{idsuc nomcli} \rightarrow \text{nombanq}$
 - La violación de esta df no se detectará a menos que se calcule la reunión (JOIN) entre $R1$ y $R2$
- ▶ No se puede llevar a la FNBC

8.5 Enfoques de diseño relacional

Síntesis relacional: diseño ascendente

- Obtiene distintos **esquemas de relación R_i** a partir de
 - Atributos y
 - Dependencias funcionales entre dichos atributos
- Agrupa información referente a un mismo concepto
- **Aplicando síntesis siempre** es posible encontrar una **descomposición de reunión sin pérdida** que preserve las dependencias y esté en **3FN**

8.5 Enfoques de diseño relacional: 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 α
3. Formar una **R_i para cada grupo**, con los atributos que aparecen en el grupo y las dependencias funcionales de DF^m con α 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

8.5 Enfoques de diseño relacional: Síntesis

Ejemplo 1

- Aplicar síntesis para dejar en **3FN** el siguiente esquema de relación
BANQUERO_PERSONAL(idsuc, nomcli, nombanq, idoficina)
donde $DF^m = \{ \text{nombanq} \rightarrow \text{idsuc idoficina}, \text{idsuc nomcli} \rightarrow \text{nombanq} \}$
y las claves $K = \{ (\text{idsuc}, \text{nomcli}), (\text{nomcli}, \text{nombanq}) \}$
- paso 1. ya tenemos DF^m
- pasos 2 y 3
 $R1(\text{nombanq}, \text{idsuc}, \text{idoficina}); DF1 = \{ \text{nombanq} \rightarrow \text{idsuc idoficina} \}; K1 = (\text{nombanq})$
 $R2(\text{idsuc}, \text{nomcli}, \text{nombanq}); DF2 = \{ \text{idsuc nomcli} \rightarrow \text{nombanq} \}; K2 = (\text{idsuc}, \text{nomcli})$
- paso 4
Todos los atributos han sido colocados
- paso 5
 $R2$ ya contiene una clave de BANQUERO_PERSONAL, luego el proceso acaba
- El conjunto de esquemas resultante es $\{ R1, R2 \}$, todos están en **3FN**

8.5 Enfoques de diseño relacional: Síntesis

Ejemplo 2

$R(a, b, c, d, e, f, g) \quad K = (b, e, g) \quad DF^m = \{ b \rightarrow a, c, e \rightarrow d, d \rightarrow f \}$

Grupos:

$G1 = \{ b \rightarrow a, c \}$

$G2 = \{ e \rightarrow d \}$

$G3 = \{ d \rightarrow f \}$

Relaciones:

$R1(b, a, c) \quad K1 = (b) \quad DF1 = \{ b \rightarrow a, c \}$

$R2(e, d) \quad K2 = (e) \quad DF2 = \{ e \rightarrow d \}$

$R3(d, f) \quad K3 = (d) \quad DF3 = \{ d \rightarrow f \}$

El atributo 'g' no está en ninguna relación:

$R4(g) \quad K4 = (g) \quad DF4 = \emptyset$

La clave no está en ninguna relación

$R5(b, e, g) \quad K5 = (b, e, g) \quad DF5 = \emptyset$

Obsérvese que no se obtiene exactamente el mismo resultado que mediante el proceso de análisis (ejemplo 2)

Conjunto de esquemas resultado: $R1, R2, R3, R4, R5$ (todas en 3FN (y FNBC))

8.5 Enfoques de diseño relacional

Comparación análisis vs. síntesis

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

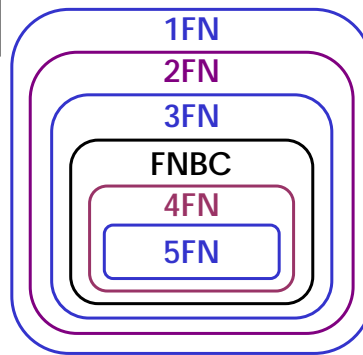
8.5 Enfoques de diseño relacional

Para terminar este apartado...

- Lo más correcto es tener los **esquemas de relación 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 la 3FN o a la FNBC
- El proceso de normalización **disminuye la redundancia**
 - Por tanto consigue que las relaciones sean menos vulnerables a las propiedades no deseables
- ... pero **penaliza las consultas**
 - Pues será necesario realizar JOIN, cuya ejecución suele ser costosa, entre relaciones resultado de la descomposición

Para terminar el tema...

La relación entre las diversas formas normales



PROPIEDADES	3FN	FNBC	4FN
Elimina redundancia debida a las dependencias funcionales	la mayor parte	sí	sí
Conserva las dependencias funcionales	sí	quizás	quizás
Elimina la redundancia debida a las dependencias multivaloradas	no	no	sí
Conserva las dependencias multivaloradas	quizás	quizás	quizás

8.6 Dependencias multivaloradas y 4FN

- Es posible que todavía exista redundancia en un esquema de relación en FNBC
- Una **dependencia multivalorada**, dmv, proviene de **dos o más atributos multivalorados** que son **independientes** entre sí
 - ESTUDIANTE_RESIDENTE(dni, estudios, actividad)
 - ACTOR(nomactor, calle, ciudad, titulopeli, añoopeli)
- La única forma de expresar la **independencia** de los conjuntos de valores ...
 - estudios vs. actividades de los estudiantes
 - direcciones de los actores vs. películas en las que han participado
 ... es hacer que
 - para cada actor, cada dirección aparezca con cada película, y
 - para cada estudiante, cada valor de estudios aparezca con cada actividad
- Las dmv son una consecuencia de la transformación de una R en la 1FN, cuando existen varios atributos multivalorados e independientes entre sí

NO
ENTRA

8.6 Dependencias multivaloradas y 4FN

Definición formal de dependencia multivalorada

- Sea el esquema de relación $R(A, D)$ y dos descriptores α y β ,

$\alpha \twoheadrightarrow \beta$ (α **multidetermina** β) si y sólo si,
para cada par de tuplas t_1 y t_2 de $r(R)$ tales que $t_1[\alpha] = t_2[\alpha]$
es posible encontrar un par de tuplas t_3 y t_4 en $r(R)$ que cumplen

NO
ENTRA

- $t_1[\alpha] = t_2[\alpha] = t_3[\alpha] = t_4[\alpha]$
- $t_3[\beta] = t_1[\beta]$ y $t_4[\beta] = t_2[\beta]$
- $t_3[A - \alpha - \beta] = t_2[A - \alpha - \beta]$ y $t_4[A - \alpha - \beta] = t_1[A - \alpha - \beta]$

- Para cada valor de α , existen 0 o más valores de β , independientemente de los valores del resto de atributos ($A - \alpha - \beta$)
- Que se verifique $\alpha \twoheadrightarrow \beta$ sólo depende de α y β , pero que se cumpla $\alpha \twoheadrightarrow \beta$ depende además del resto de atributos: las dmvs dependen del contexto
- $\alpha \twoheadrightarrow \beta$ significa que **siempre que aparezcan dos tuplas con distintos valores de β , pero el mismo en α , los valores de β deberán repetirse con cada valor distinto de $A - \alpha - \beta$ que ocurra con dicho valor de α**
- Ejemplo: nombreactor \twoheadrightarrow calle, ciudad

8.6 Dependencias multivaloradas y 4FN

Reglas de inferencia para dmvs

- **Regla de dependencias triviales**

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

NO
ENTRA

- $\alpha \twoheadrightarrow \gamma$, donde $\gamma \subseteq \beta$ pues se han quitado componentes de β que también estaban en α
- $\alpha \twoheadrightarrow \delta$, donde $\beta \subseteq \delta$ pues se han añadido componentes a β que también estaban en α , 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 de transitividad para dmv**

Si R cumple $\alpha \twoheadrightarrow \beta$ y $\beta \twoheadrightarrow \gamma$, entonces cumple $\alpha \twoheadrightarrow \gamma - \beta$

- **Regla de replicación**

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

8.6 Dependencias multivaloradas y 4FN

Reglas de inferencia para dmvs (y 2). Dmv trivial

- **Regla de complemento**

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

NO
ENTRA

- **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 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 b_i (es decir $\alpha \cup \beta = A$)

- **Ejemplo de dmv no trivial** nombreactor \twoheadrightarrow calle, ciudad

8.6 Dependencias multivaloradas y 4FN

Cuarta forma normal

NO
ENTRA

- Es una generalización de la FNBC, que elimina las dmv no triviales
- R(A, D) está en **4FN** si **para toda dmv no trivial $\alpha \twoheadrightarrow \beta$, se cumple que α es una superclave de R**

Descomposición en la 4FN

- Si R no está en 4FN, por contener una dmv $\alpha \twoheadrightarrow \beta$, donde α no contiene una clave de R, se debe **dividir** R en 2 esquemas:
 - R1, con todos los atributos de α y los de β
 - R2, con todos los atributos de α y el resto de atributos $A-\alpha-\beta$
 - » Asignar adecuadamente a cada Ri las df y dmv
 - » Cumple los principios de **descomposición sin pérdidas** puesto que $\alpha \twoheadrightarrow \beta$ y $\alpha \twoheadrightarrow A-\alpha-\beta$
 - R1(nombreactor, calle, ciudad)
 - R2(nombreactor, titulopeli, añopeli)