



Dpto. Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur

ELEMENTOS DE BASES DE DATOS

Segundo Cuatrimestre 2015

Clase 10:

**Diseño Relacional – Formas
Normales y Descomposiciones
(Parte III)**

Mg. María Mercedes Vitturini
[mvitturi@uns.edu.ar]



Forma Normal

La **forma normal** es una condición sobre un esquema de **base de datos** en función de los esquemas de relación, sus dependencias funcionales y sus llaves candidatas.

- Las formas normales que vamos a estudiar son:
 - **Primer Forma Normal (1FN),**
 - **Segunda Forma Normal (2FN),**
 - **Tercer Forma Normal (3FN), y**
 - **Forma Normal de Boyce Codd (FNBC).**



Normalización

La **normalización** es el proceso por el que sucesivamente se reducen esquemas de relación con anomalías para producir nuevos esquemas mejor estructurados.

✓ Objetivos de la normalización

- 👍 Minimizar la redundancia de datos.
 - 👍 Simplificar la gestión de los datos.
 - 👍 Proveer un diseño que favorece el crecimiento de la información.
- **En general, una base de datos normalizada no favorece:**
 - La simplicidad de las consultas.
 - El tiempo de acceso a la información que está dispersa.

Conceptos previos

- Atributos: [dominio atómico](#).
- Atributos: [Primos y No Primos](#).
- [Dependencia funcional total y parcial](#).

Formas normales

- [1FN](#)
 - Definición y ejemplos
- [2FN](#)
 - Definición y ejemplos
- [3FN](#)
 - Definición y ejemplos
- [FNBC](#)
 - Definición y ejemplos

Conceptos

El proceso de normalización

- [Cuestiones previas](#).

Descomposiciones

- [Descomponer para alcanzar la 1FN](#)
- [Descomponer para alcanzar la 2FN](#)
- [Descomponer para alcanzar la 3FN](#)
- [Descomponer para alcanzar la FNBC](#)
- [Ejemplo](#)

Primera Forma Normal (1FN)

Definición: un esquema de relación R está en **primera forma normal (1FN)** si los valores para $\text{dom}(A)$ son atómicos para todo atributo A en R.

- Un *dominio es atómico* si sus elementos se pueden considerar como unidades indivisibles.
 - Exige que los valores para los dominios no sean valores compuestos (conjuntos, listas, arreglos, etc.).
- Características de la 1FN:
 - Admite representación directa en tablas.
 - Lenguajes de consulta más simples.
 - Es posible definir restricciones o dependencias funcionales entre atributos.

Ejemplos – No satisfacen 1FN

Los siguientes esquemas no cumplen con la 1FN:

- 1. Personas = (DNI, Apellido, Nombres, Domicilio, $\{Hobby\}_n$)**
 - El grupo que se repite de hobby **no respeta la 1FN.**
- 2. Artículos = (código, nombre, descripción, nombDpto)**
 - Con *dom (código)* valores de la forma **dd-cccc**, donde **dd** representa el departamento que lo comercializa y **cccc** el número interno de artículo. Existe la dependencia $cccc \rightarrow dd$.
Artículos NO respeta la 1FN.

Definiciones auxiliares

Definición – Dados un esquema de relación R , un atributo A en R y un conjunto de df's F , un atributo A es **primo en R con respecto a F** si *A es parte de alguna llave candidata de R* . De lo contrario, se dice que A es **no primo**.

Definición – Dados un conjunto de df's F y una df $X \rightarrow Y$, se dice que Y es **parcialmente dependiente de X** si en F , $X \rightarrow Y$ no está reducido a izquierda, esto es, *Y depende de un subconjunto de los atributos de X* . Si $X \rightarrow Y$ está reducido a izquierda entonces *Y es totalmente dependiente de X* .

Segunda Forma Normal (2FN)

Definición #1: Un esquema de relación R está en **segunda forma normal (2FN)** con respecto a un conjunto de df's F si está en 1FN y cada atributo *no primo* es totalmente dependiente de cada llave candidata en R.

Definición #2: Un esquema de relación R está en **segunda forma normal (2FN)** con respecto a un conjunto de df's F si para cada df $X \rightarrow A$ ($A \notin X$) en F se verifica que:

- X no es un subconjunto propio de una llave, o bien:
- A es primo.

2FN admite df
entre no primos

Ejemplo

ENTRADAS (funNro, fecha, espectador, asientoNro)

(en forma reducida $R(N, F, E, A)$)

- Para ENTRADAS se especifican las siguientes df's:

$$F = \{ NE \rightarrow A, N \rightarrow F \}$$

- **Llave:** NE, con
 - **N** y **E** son atributos **primos**,
 - **A** y **F** son **no primos**.

✘ ENTRADAS no respeta 2FN.

- Justificación: el atributo no primo *fecha* depende parcialmente de la llave NE, dado que existe la df **N → F en F**.

Problemas del esquema

ENTRADAS (funNro, fecha, espectador, asientoNro)

- *Entradas* tiene los siguientes problemas de diseño:
 - Se repite la información número de función con su fecha en tantas tuplas como espectadores tenga asociados una función.
 - Para agregar una función en la relación *entradas* se necesita tener al menos un espectador.
 - Si el único espectador de una función se borra, se pierde información sobre la fecha de la función.
 - En caso de cambiar la fecha de una función se debería cambiar en todos los espectadores asociados a la misma.

Una mejor organización

- Para R (NFEA) y $F = \{ NE \rightarrow A, N \rightarrow F \}$.

- La descomposición:

$\rho = (R_1, R_2)$ donde R_1 (NEA) y R_2 (NF)

- $R_1 = (NEA)$, $F_1 = \{ NE \rightarrow A \}$ llave NE, satisface la 2FN.
 - $R_2 = (NF)$, $F_2 = \{ N \rightarrow F \}$ llave N, satisface la 2FN.
- Dado que cada subesquema satisface la 2FN, luego la descomposición ρ satisface la 2FN.
 - ¿Qué puede decir de esta descomposición con relación a las propiedades JSP y PD?

Ejercicios

- Dados los siguientes esquemas y sus df's, determinar si satisfacen 2FN:

a. **Libros**(ISBN, Título, Nombre_Autor) \equiv L(I,T,A)

$$F = \{ I \rightarrow T \}$$

b. **Cientes** = (CliNro, CliNombre, CliDomicilio, CódPostal, NomLocalidad) \equiv C (C,N,D,P,L)

$$F = \{ C \rightarrow NDP, P \rightarrow L \}$$

Tercera Forma Normal (3FN)

Definición #1: Un esquema de relación R está en **tercera forma normal (3FN)** con respecto a un conjunto de df's F si está en 1FN y cada atributo no primo NO es transitivamente dependiente de una superllave de R.

Definición #2: Un esquema de relación R está en **tercera forma normal (3FN)** si para cada dependencia funcional $X \rightarrow A$ ($A \notin X$) en F se verifica que:

- X es llave, o bien,
- A es primo.

3FN admite df entre primos

Ejemplo

Dado:

VUELOS (VueloNro, Fecha, PilotoNro, NombPiloto)

$F = \{VF \rightarrow P, P \rightarrow N\}$

- Llave **VF** con:
 - **V, F** atributos **primos** y
 - **P, N** son **no primos**.
 - **VUELOS respeta 2FN, pero no respeta en 3FN.**
- La descomposición de R:
 $\rho = (R_1, R_2)$ con $R_1(VFP)$ $R_2(PN)$
- y las df's $\{VF \rightarrow P\}$ en R_1 y $\{P \rightarrow N\}$ en R_2 satisface 3FN.
- ¿Es JSP y PD?

Problemas de Diseño

VUELOS (VueloNro, Fecha, PilotoNro, NombPiloto)

- Vuelos tiene los siguientes problemas de diseño:
 - Sólo se posible mantener información de un piloto si está asociado a un vuelo.
 - Se mantiene la información de un piloto y su nombre tantas veces como vuelos tenga asignados.
 - Si fuese necesario modificar el nombre del piloto, debería corregirse en todos los vuelos a los que fue asignado.

Forma Normal de Boyce-Codd (FNBC)

Definición: Un esquema de relación R está en **forma normal de Boyce-Codd (FNBC)** con respecto a un conjunto de df's F si para cada dependencia funcional $X \rightarrow A$ ($A \notin X$, esto es no trivial) se verifica que X es llave.

• Ejemplo:

– **Cientes = (ClienteNro, Nombre, Domicilio, FechaNac)**

≡ **C (C, N, D, F)**

– $F = \{C \rightarrow N, D, F\}$

FNBC la única df no trivial es de la llave

Descomposiciones normalizadas

- Los esquemas de relación (almacenamientos) que no respetan por lo menos la **3FN** *no responden a un buen diseño relacional de datos*. Tienen problemas de diseño que se traducen en:
 - Redundancia de información.
 - Inconsistencias.
 - Anomalías de inserción y de borrado.
- La solución a un problema de diseño de datos es encontrar una descomposición mejor.

Tip's

- Cuando un diseño relacional no es de calidad, entonces se busca **descomponerlo en dos o más esquemas**.
- Se espera que cada uno de los **sub-esquemas respeten una buena forma normal (FNBC ó 3FN)**.
- El criterio de descomposición lo fija el conjunto de dependencias funcionales. Por ello, se parte de un **conjunto de dependencias funcionales o restricciones mínimo y reducido**.
- Vamos a ver una **forma intuitiva de descomposición**.
- Esta forma intuitiva está basada en conceptos formales de la teoría relacional

Descomposiciones normalizadas

Supongamos la siguiente definición para un esquema de relación *Dicta*, que mantiene información de los docentes y las materias que dictan cada año y cuatrimestre:

DICTA (prof-legajo, prof-apellido, prof-nombre, prof-domicilio,
_0{mat-código, dicta-año, dicta-cuat, mat-nombre, dpto-código,
dpto-nombre}_n)

F = {prof-legajo → prof-apellido, prof-nombre, prof-domicilio;
mat-código → mat-nombre, dpto-código;
dpto-código → dpto-nombre}

Primer Forma Normal

- El esquema **DICTA** NO RESPETA LA 1FN:
 - *Existen atributos no atómicos.*
- En este caso por cada docente se mantiene un grupo no atómico con las materias que el profesor dicta y /o ha dictado en distintos cuatrimestres.
- Para contar con un modelo de datos que respete la **1FN**, la solución es descomponer.

Descomposición en 1FN

- Si un esquema no respeta la 1FN se debe descomponer en dos esquemas:
 1. Un esquema con todos los datos que no pertenecen al grupo repetitivo. Se identifica el/los atributos que son clave de este esquema.
 2. Un esquema con el grupo que se repite al que se le agrega la clave del esquema del inciso anterior. Definir la clave para este nuevo esquema.
 3. Se identifican las df's de cada esquema.

Descomposición en 1FN

- **Dicta₁** (prof-legajo, prof-apellido, prof-nombre, prof-domicilio).
- **Dicta₂** (prof-legajo, mat-código, dicta-año, dicta-cuat,
mat-nombre, dpto-código, dpto-nombre)

Esquema Dicta₁

- **Dicta₁** (prof legajo, prof-apellido, prof-nombre, prof-domicilio)
 - **Dependencias Funcionales:**
 - $F_1 = \{\text{prof-legajo} \rightarrow \text{prof-apellido}, \text{prof-nombre}, \text{prof-domicilio}\}$
 - **Clave:** prof-legajo.
 - **Atributos primos:** prof-legajo.
 - **Atributos no primos:** prof-apellido, prof-nombre, prof-domicilio.

Esquema Dicta₂

- **Dicta₂** (prof-legajo, mat-código, dicta-año, dicta-cuat, mat-nombre, dpto-código, dpto-nombre)
 - **Dependencias Funcionales:**
 - $F_2 = \{\text{mat-código} \rightarrow \text{mat-nombre, dpto-código}; \text{dpto-código} \rightarrow \text{dpto-nombre}\}$
 - **Clave:** prof-legajo + mat-código + dicta-cuat + dicta-año.
 - **Atributos primos:** prof-legajo, mat-código, dicta-cuat, dicta-año
 - **Atributos no primos:** mat-nombre, dpto-código, dpto-nombre

Segunda Forma Normal

- **DICTA₁** respeta la **2FN** ya que la clave esta formada por un único atributo.
 - **DICTA₁** (prof-legajo, prof-apellido, prof-nombre, prof-domicilio).

Segunda Forma Normal

DICTA₂ NO RESPETA la 2FN.

DICTA₂ (prof-legajo, mat-código, dicta-cuat, dicta-año
mat-nombre, dpto-código, dpto-nombre)

- **Clave:** prof-legajo, mat-código, dicta-cuat, dicta-año.
- Existe una dependencia funcional:
 - **mat-código → mat-nombre, dpto-código, dpto-nombre**
 - *Los atributos no primos:* mat-nombre, dpto-código y dpto-nombre *dependen parcialmente de la llave.*

Descomposición en 2FN

- La solución para un almacenamiento que no respeta la **2FN** es descomponer:
 1. Un esquema de relación con los atributos clave primaria y aquellos atributos que dependen totalmente de la clave primaria (si los hubiera). La clave es la clave original.
 2. Un esquema de relación con cada parte de la clave y los atributos que dependen de esa parte de la clave, directa y/o transitivamente. La clave primaria para este esquema es el conjunto de atributos parcial de la clave original.
 3. Se identifica el conjunto de df's de cada esquema.

Descomposición en 2FN

- **Dicta₂** = (prof-legajo, mat-código, dicta-cuat, dicta-año, mat-nombre, dpto-código, dpto-nombre)
 - **Dicta₂₁** (prof-legajo, mat-código, dicta-cuat, dicta-año)
 - **Dicta₂₂** (mat-código, mat-nombre, dpto-código, dpto-nombre).

Esquema Dicta₂₁

Dicta₂₁ (prof-legajo, mat-código, dicta-cuat, dicta-año)

- **Dependencias Funcionales:** no posee.
- **Clave:** prof-legajo + mat-código + dicta-cuat + dicta-año.
- **Atributos primos:** todos.
- **Atributos no primos:** no posee.

Esquema Dicta₂₂

Dicta₂₂(mat-código, mat-nombre, dpto-código, dpto-nombre)

- **Dependencias Funcionales:**

- $F_{22} = \{\text{mat-código} \rightarrow \text{mat-nombre}, \text{dpto-código} ; \text{dpto-código} \rightarrow \text{dpto-nombre}\}$

- **Clave:** mat-código.

- **Atributos primos:** mat-código.

- **Atributos no primos:** mat-nombre, dpto-código, dpto-nombre.

Tercer Forma Normal

- El almacenamiento **DICTA₂₁** respeta la **3FN** ya que no posee ningún atributo que no forme parte de la clave, por lo tanto no existen dependencias entre atributos no primos.

DICTA₂₁(prof legajo,mat código,dicta año,dicta cuat)

Tercer Forma Normal

DICTA₂₂ NO RESPETA 3FN.

- Existe una dependencia funcional entre los atributos no primos *dpto-código y dpto-nombre*.
- Ninguno de ellos forma parte de la clave *mat-código*.
- La solución está en descomponer al almacenamiento

DICTA₂₂

Descomposición 3FN

- Si *un esquema no respeta la 3FN, corresponde descomponerlo* en dos esquemas.
 1. Un esquema de relación con la clave primaria y los atributos que dependen totalmente de la clave primaria (sin incluir los que dependen transitivamente). La clave de este esquema es la clave del esquema origen
 2. Otro esquema de relación con los atributos participan de la dependencia transitiva. La clave será los atributos izquierdos de la dependencia transitiva.
- En el ejemplo:

Descomposición en 3FN

- **Dicta22** (mat-código, mat-nombre, dpto-código, dpto-nombre)
- **Dicta₂₂₁** (mat-código, mat-nombre, dpto-código)
- **Dicta₂₂₂** (dpto-código, dpto-nombre)

Esquema Dicta₂₂₁

Dicta₂₂₁ (mat-código, mat-nombre, dpto-código)

- **Dependencias Funcionales:**

- $F_{221} = \{ \text{mat-código} \rightarrow \text{mat-nombre, dpto-código} \}$

- **Clave:** mat-código

- **Atributos primos:** mat-código

- **Atributos no primos:** mat-nombre, dpto-código.

Esquema Dicta₂₂₂

Dicta₂₂₂ (dpto-código, dpto-nombre)

- **Dependencias Funcionales:**
 - $F_{222} = \{\text{dpto-código} \rightarrow \text{dpto-nombre}\}$
- **Clave:** dpto-código.
- **Atributos primos:** dpto-código
- **Atributos no primos:** dpto-nombre.

Descomposición en 3FN

ρ (DICTA_1 , DICTA_{21} , DICTA_{221} , DICTA_{222}) donde

- DICTA_1 (prof-legajo, prof-apellido, prof-nombre, prof-domicilio)
 $F_1 = \{\text{prof-legajo} \rightarrow \text{prof-apellido}, \text{prof-nombre}, \text{prof-domicilio}\}$
- DICTA_{21} (prof-legajo, mat-código, dicta-año, dicta-cuat)
 $F_{21} = \{\emptyset\}$
- DICTA_{221} (mat-código, mat-nombre, dpto-código)
 $F_{221} = \{\text{mat-código} \rightarrow \text{mat-nombre}, \text{dpto-código}\}$
- DICTA_{222} (dpto-código, dpto-nombre)
 $F_{222} = \{\text{dpto-código} \rightarrow \text{dpto-nombre}\}$

Descomposición en 3FN con nombres significativos

- **PROFESOR** = (prof-legajo, prof-apellido, prof-nombre, prof-domicilio)
- **DICTA** = (prof-legajo, mat-código, dicta-año, dicta-cuat)
- **MATERIA** = (mat-código, mat-nombre, dpto-código)
- **DEPARTAMENTO** = (dpto-código, dpto-nombre)

ρ (**PROFESOR** , **DICTA** , **MATERIA**, **DEPARTAMENTO**)

En el ejemplo:

ρ (**PROFESOR , DICTA , MATERIA, DEPARTAMENTO**)

cada subesquema R_i tiene como única dependencia funcional no trivial $X \rightarrow Y$, con X llave para R_i , por lo tanto ρ está en FNBC.

- Dado un esquema R , si R no está en una buena forma normal (3FN o FNBC):
 - *Siempre* es posible encontrar una descomposición **3FN, pd, jsp**.
 - *Siempre* es posible encontrar una descomposición **FNBC, jsp**.

¿Cómo se diseña el modelo de datos?

- Diseñar el modelado conceptual
 - Modelo Entidad Relación (MER)
 - El Diagrama Entidad Relación (DER)

*Elementos básicos:
conjuntos de entidades y
conjuntos de relaciones.*

El traslado de un DER al MR se hace siguiendo reglas

- Diseñar el modelo lógico
 - Modelo Relacional (MR)
- Revisar el diseño del MR
 - Normalizar (descomponer)

*Elemento básico: esquemas de relaciones.
Los esquemas con sus dependencias funcionales se analizan para ver que respeten al menos 3FN sino se descomponen.*

Evolución en el diseño de modelos

Temas de la clase de hoy

- Conceptos de Normalización de almacenamientos
 - Dependencias funcionales.
 - Claves de almacenamiento.
 - Atributos primos y no primos.
- Formas normales (1FN, 2FN, 3FN y FNBC)
- **Bibliografía:**
 - *Database System Concepts* – Abraham Silberschatz – Capítulo 8
 - *DataBase System – The Complete Book* – H. Molina, J. Ullman. Capítulo 3.

Formas Normales

Definición 1FN: Un esquema de relación R está en primera forma normal (1FN) si los valores en $dom(A)$ son atómicos para cada atributo A en R .

Definición 2FN (1): Un esquema de relación R está en segunda forma normal (2FN) con respecto a un conjunto de df's F si está en 1FN y cada atributo no primo es totalmente dependiente de cada llave en R .

Definición 2FN (2): Un esquema de relación R está en segunda forma normal (2FN) con respecto a un conjunto de df's F si para cada $X \rightarrow A$ ($A \notin X$) en F se verifica que:

- X no es un subconjunto propio de una llave, o bien:
- A es primo.

Definición 3FN (1): Un esquema de relación R está en tercera forma normal (3FN) con respecto a un conjunto de df's F si está en 1FN y cada atributo no primo NO es transitivamente dependiente de una superllave de R .

Definición 3FN (2): Un esquema de relación R está en tercera forma normal (3FN) con respecto a un conjunto de df's F si para cada $X \rightarrow A$ ($A \notin X$) en F se verifica que:

- X es superllave, o bien:
- A es primo.

Definición FNBC: Un esquema de relación R está en forma normal de Boyce-Codd (FNBC) con respecto a un conjunto de df's F si para cada $X \rightarrow A$ ($A \notin X$) en F se verifica que X es superllave.