

Teoría del Diseño Relacional

Parte I: Formas Normales

Alberto Fasce, Mariano Beiró

Dpto. de Computación - Facultad de Ingeniería (UBA)

Topics

- 1 Introducción
- 2 Dependencias funcionales
- 3 Formas normales: 1FN, 2FN, 3FN, FNBC
- 4 Dependencias multivaluadas y 4FN
- 5 Dependencias de junta y 5FN
- 6 Bibliografía

1 Introducción

2 Dependencias funcionales

3 Formas normales: 1FN, 2FN, 3FN, FNBC

4 Dependencias multivaluadas y 4FN

5 Dependencias de junta y 5FN

6 Bibliografía

Introducción

- Criterios de un buen diseño relacional
 - Preservación de información
 - Redundancia mínima
- Cuando se parte de un correcto diseño conceptual y se hace un correcto pasaje al modelo lógico, se obtiene un esquema sin redundancia y se preserva toda la información del mundo real que se quería modelar.
- Pero, ¿cómo verificamos un esquema relacional? ¿cómo corregimos un esquema que fue mal diseñado?
- La **teoría del diseño relacional** formaliza estos requisitos a través de las **formas normales**.

1 Introducción

2 Dependencias funcionales

3 Formas normales: 1FN, 2FN, 3FN, FNBC

4 Dependencias multivaluadas y 4FN

5 Dependencias de junta y 5FN

6 Bibliografía

Dependencia funcional

Definición

- Dada una relación $R(\bar{A})$, una **dependencia funcional** $X \rightarrow Y$, con $X, Y \subset \bar{A}$ es una restricción sobre las posibles tuplas de R que implica que dos tuplas con igual valor del conjunto de atributos X deben también tener igual valor del conjunto de atributos Y .
- Ésto es:

$$\forall s, t \in R : s[X] = t[X] \rightarrow s[Y] = t[Y]$$

- La dependencia funcional $X \rightarrow Y$ implica que hay una relación funcional entre los valores de X y los de Y dentro de la base de datos.
- Cuando $Y \subset X$ decimos que $X \rightarrow Y$ es **trivial**.
- Las dependencias funcionales se definen a partir de la semántica de los datos. ¡No es posible inferirlas viendo los datos!

- 1 Introducción
- 2 Dependencias funcionales
- 3 Formas normales: 1FN, 2FN, 3FN, FNBC**
- 4 Dependencias multivaluadas y 4FN
- 5 Dependencias de junta y 5FN
- 6 Bibliografía

Formas normales

- Las **formas normales** son una serie de estructuras con las que un esquema de base de datos puede cumplir ó no.
- Las formas normales clásicas son:
 - Primera forma normal (**1FN**) (E. Codd, 1970)
 - Segunda forma normal (**2FN**) (E. Codd, 1971)
 - Tercera forma normal (**3FN**) (E. Codd, 1971)
 - Forma normal Boyce-Codd (**FNBC**) (R. Boyce E. Codd, 1974)
 - Cuarta forma normal (**4FN**) (R. Fagin, 1977)
 - Quinta forma normal (**5FN**) (R. Fagin, 1979)
- Cada forma normal es más fuerte que las anteriores —en el orden en que las hemos introducido—. Entonces:

$S \text{ está en } 5FN \rightarrow S \text{ está en } 4FN \rightarrow \dots S \text{ está en } 2FN \rightarrow S \text{ está en } 1FN.$

Formas normales

- En 1972 E. Codd propuso el concepto de **normalización** como el proceso a través del cual se convierte un esquema de base de datos en uno equivalente (i.e., **que preserva toda la información**) y que cumple con una determinada forma normal.
- El objetivo es:
 - Preservar la información
 - Eliminar la redundancia
 - Evitar las anomalías de ABM
- Partiremos de un conjunto de dependencias funcionales que supondremos definido por el **diseñador de la base de datos**.

Primera forma normal (1FN)

Definición

- Decimos que un esquema de base de datos relacional está en **primera forma normal (1FN)** cuando los dominios de todos sus atributos sólo permiten valores atómicos (es decir, indivisibles) y monovaluados.
- Actualmente, se considera que en el modelo relacional todos los atributos deben ser monovaluados y atómicos.
- Con este criterio, todo esquema relacional está ya en 1FN.
- Pero, ¿cómo lo resolveríamos si éste no fuera el caso?

nombre_profesor	mail
Juan Gómez	{jgomez@udbc.com.ar, jgomez94@mibase.com}
Roberta Casas	{rcasas@udbc.com.ar, rcasas@ggmail.com}
Irene Adler	{iadler@udbc.com.ar}

Primera forma normal (1FN)

Situación

■ Situación:

nombre_profesor	mail
Juan Gómez	{jgomez@udbc.com.ar, jgomez94@mibase.com}
Roberta Casas	{rcasas@udbc.com.ar, rcasas@gmail.com}
Irene Adler	{iadler@udbc.com.ar}

- Solución 1: Colocar un mail por tupla y repetir el nombre del profesor.

nombre_profesor	mail
Juan Gómez	jgomez@udbc.com.ar
Juan Gómez	jgomez94@mibase.com
Roberta Casas	rcasas@udbc.com.ar
Roberta Casas	rcasas@gmail.com
Irene Adler	iadler@udbc.com.ar

- Solución 2: Suponer un máximo posible M de mails y tener M atributos distintos reservados a tal fin. Para profesores que tienen menos de M mails, quedarán valores nulos.

nombre_profesor	mail1	mail2
Juan Gómez	jgomez@udbc.com.ar	jgomez94@mibase.com
Roberta Casas	rcasas@udbc.com.ar	rcasas@gmail.com
Irene Adler	iadler@udbc.com.ar	NULL

Segunda forma normal (2FN)

Situación

- Consideremos ahora el siguiente ejemplo, ya en 1FN:

nombre_dpto	nombre_profesor	asignatura
Física	Juan Gómez	Física II
Física	Roberta Casas	Física II
Física	Juan Gómez	Física III
Matemática	Roberta Casas	Topología
Matemática	Irene Adler	Álgebra I

- Identifiquemos las dependencias funcionales semánticas:
 - *asignatura* → *nombre_dpto*
- Existen otras dependencias funcionales que pueden deducirse de la anterior:
 - { *nombre_profesor*, *asignatura* } → *nombre_dpto*
- Y otras que son **triviales**:
 - Ejemplo: { *nombre_profesor*, *asignatura* } → *asignatura*

Segunda forma normal (2FN)

Situación

nombre_dpto	nombre_profesor	asignatura
Física	Juan Gómez	Física II
Física	Roberta Casas	Física II
Física	Juan Gómez	Física III
Matemática	Roberta Casas	Topología
Matemática	Irene Adler	Álgebra I

- Identifiquemos ahora las claves candidatas de la relación:
 - CK = {nombre_profesor, asignatura}
- Ésta es la única clave candidata, y por lo tanto será la clave primaria.
- Observemos que *nombre_dpto* no depende de la clave primaria completa, sino sólo de una parte. Decimos que la dependencia $PK \rightarrow nombre_dpto$ es una **dependencia funcional parcial**.

Segunda forma normal (2FN)

Dependencia funcional parcial: Definición

- Una **dependencia funcional** $X \rightarrow Y$ es **parcial** cuando existe un subconjunto propio $A \subset X$, $A \neq X$ para el cual $A \rightarrow Y$.
- Una **dependencia funcional** $X \rightarrow Y$ es **completa** si y sólo si no es parcial.

nombre_dpto	nombre_profesor	asignatura
Física	Juan Gómez	Física II
Física	Roberta Casas	Física II
Física	Juan Gómez	Física III
Matemática	Roberta Casas	Topología
Matemática	Irene Adler	Álgebra I

- En el ejemplo, nombre_dpto no tiene dependencia funcional completa de la clave primaria {nombre_profesor, asignatura}.

Segunda forma normal (2FN)

Definición

- Atributo primo de una relación: Es aquel que es parte de alguna clave candidata de la relación.
- Decimos que una relación está en **segunda forma normal (2FN)** cuando todos sus atributos no primos tienen dependencia funcional completa de las claves candidatas.

nombre_dpto	nombre_profesor	asignatura
Física	Juan Gómez	Física II
Física	Roberta Casas	Física II
Física	Juan Gómez	Física III
Matemática	Roberta Casas	Topología
Matemática	Irene Adler	Álgebra I

- ¿Cómo resolvemos la situación en el ejemplo?
 - DocenteAsignatura(nombre_profesor, asignatura)
 - AsignaturaDepartamento(asignatura, nombre_dpto)

Segunda forma normal (2FN)

Ejemplo: Base de datos de torneos de tenis individual

TENIS									
nombre_torneo	año	ciudad	país	tenista1	tenista2	ronda	set	punt1	punt2
Roland Garros	2016	París	Francia	A. Murray	S. Wawrinka	2-final	1	6	4
Roland Garros	2016	París	Francia	A. Murray	S. Wawrinka	2-final	2	6	2
Roland Garros	2016	París	Francia	A. Murray	S. Wawrinka	2-final	3	4	6
Roland Garros	2016	París	Francia	A. Murray	S. Wawrinka	2-final	4	6	2
Masters de Madrid	2015	Madrid	España	R. Federer	R. Nadal	4-final	1	3	6
Masters de Madrid	2015	Madrid	España	R. Federer	R. Nadal	4-final	2	1	6
Roland Garros	2016	París	Francia	N. Djokovic	A. Murray	Final	1	6	3
Roland Garros	2016	París	Francia	N. Djokovic	A. Murray	Final	2	1	6
Roland Garros	2016	París	Francia	N. Djokovic	A. Murray	Final	3	6	2
Roland Garros	2016	París	Francia	N. Djokovic	A. Murray	Final	4	6	4

Hipótesis: Todos los torneos son por eliminación, de manera que 2 tenistas pueden enfrentarse 1 vez como máximo por torneo.

■ Identificamos las dependencias funcionales no triviales a partir de la semántica:

- $\text{nombre_torneo} \rightarrow \{\text{ciudad, país}\}$
- $\{\text{nombre_torneo, año, tenista1, tenista2}\} \rightarrow \{\text{ronda}\}$
- $\{\text{nombre_torneo, año, tenista1, ronda}\} \rightarrow \{\text{tenista2}\}$
- $\{\text{nombre_torneo, año, tenista2, ronda}\} \rightarrow \{\text{tenista1}\}$
- $\{\text{nombre_torneo, año, tenista1, tenista2, set}\} \rightarrow \{\text{punt1, punt2}\}$
- $\{\text{nombre_torneo, año, tenista1, ronda, set}\} \rightarrow \{\text{punt1, punt2}\}$
- $\{\text{nombre_torneo, año, tenista2, ronda, set}\} \rightarrow \{\text{punt1, punt2}\}$

Segunda forma normal (2FN)

Ejemplo: Base de datos de torneos de tenis individual

- Identificamos la clave primaria
 - {nombre_torneo, año, tenista1, tenista2, set}
- Identificamos otras claves candidatas
 - {nombre_torneo, año, tenista1, ronda, set}
 - {nombre_torneo, año, tenista2, ronda, set}
- ¿Dependencias funcionales parciales de una clave candidata de atributos no primos?
 - nombre_torneo → {ciudad, país}
- Descomposición:

Descomposición

Torneos(nombre_torneo, ciudad, país)

Partidos(nombre_torneo, año, tenista1, tenista2, set, ronda, punt1, punt2)

Segunda forma normal (2FN)

Ejemplo: Base de datos de torneos de tenis individual

TORNEOS

nombre_torneo	ciudad	país
Roland Garros	París	Francia
Masters de Madrid	Madrid	España

PARTIDOS

nombre_torneo	año	tenista1	tenista2	ronda	set	punt1	punt2
Roland Garros	2016	A. Murray	S. Wawrinka	2-final	1	6	4
Roland Garros	2016	A. Murray	S. Wawrinka	2-final	2	6	2
Roland Garros	2016	A. Murray	S. Wawrinka	2-final	3	4	6
Roland Garros	2016	A. Murray	S. Wawrinka	2-final	4	6	2
Masters de Madrid	2015	R. Federer	R. Nadal	4-final	1	3	6
Masters de Madrid	2015	R. Federer	R. Nadal	4-final	2	1	6
Roland Garros	2016	N. Djokovic	A. Murray	Final	1	6	3
Roland Garros	2016	N. Djokovic	A. Murray	Final	2	1	6
Roland Garros	2016	N. Djokovic	A. Murray	Final	3	6	2
Roland Garros	2016	N. Djokovic	A. Murray	Final	4	6	4

Descomposición

Definición

- En el ejemplo anterior hemos “descompuesto” la relación *Tenis* en dos relaciones: *Torneos* y *Partidos*.
- Pero, ¿qué es exactamente una descomposición de una relación?
- Partimos del concepto de **relación universal**: una relación $R(A_1, A_2, \dots, A_n)$ que engloba todos los atributos del mundo real que nuestro modelo lógico representa.
- Dada una relación universal $R(A_1, A_2, \dots, A_n)$ y un conjunto de dependencias funcionales F definidas sobre ella, decimos que un conjunto de relaciones $\{R_1(B_{11}, B_{12}, \dots, B_{1n_1}), \dots, R_m(B_{m1}, B_{m2}, \dots, B_{mn_m})\}$ es una **descomposición** de R cuando todos los atributos de la relación R se conservan. Es decir:

$$\bigcup_{i=1}^n A_i = \bigcup_{i=1}^m \bigcup_{j=1}^{n_i} B_{ij}$$

Descomposición

Propiedades

- Analizaremos dos propiedades de las descomposiciones:
 - La preservación de información.
 - La preservación de dependencias funcionales.
- Si una descomposición cumple que para toda instancia posible de R , la junta de las proyecciones sobre los R_i permite recuperar la misma instancia de relación, entonces decimos que la descomposición **preserva la información**.
- Diremos que la descomposición **preserva las dependencias funcionales** cuando toda dependencia funcional $X \rightarrow Y$ en R puede inferirse a partir de dependencias funcionales definidas en los R_i .

Descomposición

Propiedades

- Una descomposición de R que cumple con ambas propiedades se denomina **descomposición equivalente de R** .
- A medida que avanzamos en la normalización, se minimiza la **redundancia de datos**, una propiedad deseable en todo esquema de base de datos.
- En el ejemplo anterior, la descomposición en {Torneos, Partidos} es una descomposición equivalente porque preserva la información ($\text{Tenis} = \text{Torneos} * \text{Partidos}$) y preserva las dependencias funcionales. A la vez, reduce la redundancia de datos respecto a la relación *Tenis* inicial.

Tercera forma normal (3FN)

Situación

- Veamos el siguiente ejemplo:

VENTAS						
nro_factura	cliente	nro_item	cod_producto	nombre_producto	cantidad	precio_unit
0003-45821	Lionel Pessari	1	249	Suprabond 500mg	2	87.00
0003-45821	Lionel Pessari	2	230	Tersuave azul 4l	1	270.00
0003-45821	Lionel Pessari	3	115	Brocha 5cm	2	90.00
0003-45822	Claudia Serrano	1	258	Alba p/Exteriores 3l	2	225.00
0003-45822	Claudia Serrano	2	116	Brocha 10cm	2	130.00
0003-45823	Claudia Serrano	1	330	Cetol 2l	1	315.00

- Identificamos las dependencias funcionales no triviales a partir de la semántica:
 - *nro_factura* → *cliente*
 - { *nro_factura*, *nro_item* } → { *nombre_producto*, *cod_producto*, *cantidad*, *precio_unit* }
 - *cod_producto* → *nombre_producto*
- Identificamos la clave primaria:
 - { *nro_factura*, *nro_item* }
- No hay otras claves candidatas.

Tercera forma normal (3FN)

Situación

VENTAS						
nro_factura	cliente	nro_item	cod_producto	nombre_producto	cantidad	precio_unit
0003-45821	Lionel Pessari	1	249	Suprabond 500mg	2	87.00
0003-45821	Lionel Pessari	2	230	Tersuave azul 4l	1	270.00
0003-45821	Lionel Pessari	3	115	Brocha 5cm	2	90.00
0003-45822	Claudia Serrano	1	258	Alba p/Exteriores 3l	2	225.00
0003-45822	Claudia Serrano	2	116	Brocha 10cm	2	130.00
0003-45823	Claudia Serrano	1	330	Cetol 2l	1	315.00

- El esquema Ventas, ¿está en 2FN?
 - No, por la dependencia parcial de “cliente” con la clave primaria.
- Normalización:

Descomposición a 2FN

ClienteFactura(nro_factura, cliente)

DetalleFactura(nro_factura, nro_item, cod_producto, nombre_producto, cantidad, precio_unit)

Tercera forma normal (3FN)

Situación

CLIENTE FACTURA

nro_factura	cliente
0003-45821	Lionel Pessari
0003-45822	Claudia Serrano

DETALLE FACTURA

nro_factura	nro_item	cod_producto	nombre_producto	cantidad	precio_unit
0003-45821	1	249	Suprabond 500mg	2	87.00
0003-45821	2	230	Tersuave azul 4l	1	270.00
0003-45821	3	115	Brocha 5cm	2	90.00
0003-45822	1	258	Alba p/Exteriores 3l	2	225.00
0003-45822	2	116	Brocha 10cm	2	130.00
0003-45823	1	330	Cetol 2l	1	315.00

- Observemos que todas las dependencias funcionales que había se mantienen.
- Sin embargo, una de las dependencias muestra que un atributo **no primo** puede deducirse a partir de otro atributo no primo.
 - *cod_producto* → *nombre_producto*
- Entonces, decimos que *nombre_producto* tiene “dependencia transitiva” en la clave primaria, lo que no es deseable.

Tercera forma normal (3FN)

Dependencia transitiva: Definición

- Una **dependencia funcional** $X \rightarrow Y$ es **transitiva** cuando existe un conjunto de atributos Z que satisface dependencias $X \rightarrow Z$ y $Z \rightarrow Y$, siendo $Z \rightarrow Y$ no trivial, $X \rightarrow Y$ no trivial, y $Z \not\rightarrow X$.

DETALLEFACTURA

nro_factura	nro_item	cod_producto	nombre_producto	cantidad	precio_unit
0003-45821	1	249	Suprabond 500mg	2	87.00
0003-45821	2	230	Tersuave azul 4l	1	270.00
0003-45821	3	115	Brocha 5cm	2	90.00
0003-45822	1	258	Alba p/Exteriores 3l	2	225.00
0003-45822	2	116	Brocha 10cm	2	130.00
0003-45823	1	330	Cetol 2l	1	315.00

- En el ejemplo, *nombre_producto* tiene dependencia transitiva en la clave primaria porque $\{nro_factura, nro_item\} \rightarrow cod_producto$ y $cod_producto \rightarrow nombre_producto$.
- Observación:** Toda dependencia funcional parcial no trivial es transitiva.

Tercera forma normal (3FN)

Definición

- Decimos que una relación está en **tercera forma normal (3FN)** cuando no existen dependencias transitivas $CK_i \rightarrow Y$ de atributos no primos (i.e. $Y \notin \bigcup_i CK_i$), con CK_i clave candidata.
- Una definición equivalente es que para toda dependencia funcional no trivial $X \rightarrow Y$, o bien X es superclave, o bien $Y - X$ contiene sólo atributos primos.

DETALLEFACTURA					
nro_factura	nro_item	cod_producto	nombre_producto	cantidad	precio_unit
0003-45821	1	249	Suprabond 500mg	2	87.00
0003-45821	2	230	Tersuave azul 4l	1	270.00
0003-45821	3	115	Brocha 5cm	2	90.00
0003-45822	1	258	Alba p/Exteriores 3l	2	225.00
0003-45822	2	116	Brocha 10cm	2	130.00
0003-45823	1	330	Cetol 2l	1	315.00

- $PK = \{nro_factura, nro_item\}$
- ¿Cómo se resuelve la situación?
 - DetalleFactura(nro_factura nro_item cod_producto cantidad precio_unit)
 - Productos(cod_producto, nombre_producto)

Tercera forma normal (3FN)

Ejemplo: Base de datos de torneos de tenis individual

■ Volvamos al ejemplo de los tenistas:

PARTIDOS							
nombre_torneo	año	tenista1	tenista2	ronda	set	punt1	punt2
Roland Garros	2016	A. Murray	S. Wawrinka	2-final	1	6	4
Roland Garros	2016	A. Murray	S. Wawrinka	2-final	2	6	2
Roland Garros	2016	A. Murray	S. Wawrinka	2-final	3	4	6
Roland Garros	2016	A. Murray	S. Wawrinka	2-final	4	6	2
Masters de Madrid	2015	R. Federer	R. Nadal	4-final	1	3	6
Masters de Madrid	2015	R. Federer	R. Nadal	4-final	2	1	6
Roland Garros	2016	N. Djokovic	A. Murray	Final	1	6	3
Roland Garros	2016	N. Djokovic	A. Murray	Final	2	1	6
Roland Garros	2016	N. Djokovic	A. Murray	Final	3	6	2
Roland Garros	2016	N. Djokovic	A. Murray	Final	4	6	4

■ ¿Hay dependencias transitivas de atributos no primos? No.

- $\{ \text{nombre_torneo}, \text{año}, \text{tenista1}, \text{tenista2}, \text{set} \} \rightarrow \text{punt1}$
- A su vez: $\{ \text{nombre_torneo}, \text{año}, \text{tenista1}, \text{tenista2} \} \rightarrow \text{ronda}$
- Y luego: $\{ \text{nombre_torneo}, \text{año}, \text{tenista1}, \text{ronda}, \text{set} \} \rightarrow \text{punt1}$
- Pero *ronda* es parte de una clave candidata.

■ Por lo tanto, está en tercera forma normal.

Forma normal Boyce-Codd (FNBC)

Situación

- Hay un tipo de redundancia que aún no eliminamos...

CURSADA		
alumno	materia	profesor
Dante Micelli	Zoología	Edmundo Ribeiro
Dante Micelli	Botánica	José Cestoni
Dante Micelli	Anatomía General I	Pedro González
Alberto Deheza	Botánica	José Cestoni
Alberto Deheza	Zoología	Viviana Díaz
Carla Hernández	Zoología	Edmundo Ribeiro
Carla Hernández	Anatomía General I	Pedro González
Carla Hernández	Botánica	José Cestoni
Leticia Humboldt	Botánica	Héctor Larraza
Leticia Humboldt	Zoología	Viviana Díaz

Hipótesis: Cada materia es dictada por muchos profesores, pero un estudiante sólo cursa con uno de ellos. La universidad tiene la restricción de que un profesor sólo puede dictar una materia.

- Identificamos las dependencias funcionales no triviales a partir de la semántica:
 - $\{ \textit{alumno}, \textit{materia} \} \rightarrow \textit{profesor}$
 - $\textit{profesor} \rightarrow \textit{materia}$

Forma normal Boyce-Codd (FNBC)

Situación

CURSADA		
alumno	materia	profesor
Dante Micelli	Zoología	Edmundo Ribeiro
Dante Micelli	Botánica	José Cestoni
Dante Micelli	Anatomía General I	Pedro González
Alberto Deheza	Botánica	José Cestoni
Alberto Deheza	Zoología	Viviana Díaz
Carla Hernández	Zoología	Edmundo Ribeiro
Carla Hernández	Anatomía General I	Pedro González
Carla Hernández	Botánica	José Cestoni
Leticia Humboldt	Botánica	Héctor Larraza
Leticia Humboldt	Zoología	Viviana Díaz

- Identificamos la clave primaria:
 - $\{alumno, materia\}$
- Aunque hay otras claves candidatas:
 - $\{alumno, profesor\}$
- La “materia” podría deducirse con parte de la clave candidata, y sin embargo la estamos repitiendo...
- La **forma normal Boyce-Codd** impide que esto suceda prohibiendo que existan dependencias transitivas de una clave candidata, inclusive de atributos primos.
 - $\{alumno, profesor\} \rightarrow profesor \rightarrow materia$

Forma normal Boyce-Codd (FNBC)

Definición

- Una relación está en **forma normal Boyce-Codd (FNBC)** cuando no existen dependencias transitivas $CK \rightarrow Y$, con CK clave candidata.
 - Es decir, eliminamos la posibilidad de tener dependencias transitivas $X \rightarrow Y$ en las que Y es un atributo primo.
- Dicho de otra forma, una relación está en FNBC cuando para toda dependencia funcional no trivial $X \rightarrow Y$, X es superclave.
- El problema que resuelve la FNBC se da cuando en una relación existen varias claves candidatas que se solapan.
- ¿Cómo se resuelve la situación anterior?
 - Inscripciones(alumno, profesor)
 - Cursos(materia, profesor)
- ¡Pero observemos que perdimos la dependencia funcional $\{alumno, materia\} \rightarrow profesor$!

Forma normal Boyce-Codd (FNBC)

Situación

INSCRIPCIONES

alumno	profesor
Dante Micelli	Edmundo Ribeiro
Dante Micelli	José Cestoni
Dante Micelli	Pedro González
Alberto Deheza	José Cestoni
Alberto Deheza	Viviana Díaz
Carla Hernández	Edmundo Ribeiro
Carla Hernández	Pedro González
Carla Hernández	José Cestoni
Leticia Humboldt	Héctor Larraza
Leticia Humboldt	Viviana Díaz

CURSOS

materia	profesor
Zoología	Edmundo Ribeiro
Botánica	José Cestoni
Anatomía General I	Pedro González
Zoología	Viviana Díaz
Botánica	Héctor Larraza

Forma normal Boyce-Codd (FNBC)

Ejemplo: Base de datos de torneos de tenis individual

PARTIDOS							
nombre_torneo	año	tenista1	tenista2	ronda	set	punt1	punt2
Roland Garros	2016	A. Murray	S. Wawrinka	2-final	1	6	4
Roland Garros	2016	A. Murray	S. Wawrinka	2-final	2	6	2
Roland Garros	2016	A. Murray	S. Wawrinka	2-final	3	4	6
Roland Garros	2016	A. Murray	S. Wawrinka	2-final	4	6	2
Masters de Madrid	2015	R. Federer	R. Nadal	4-final	1	3	6
Masters de Madrid	2015	R. Federer	R. Nadal	4-final	2	1	6
Roland Garros	2016	N. Djokovic	A. Murray	Final	1	6	3
Roland Garros	2016	N. Djokovic	A. Murray	Final	2	1	6
Roland Garros	2016	N. Djokovic	A. Murray	Final	3	6	2
Roland Garros	2016	N. Djokovic	A. Murray	Final	4	6	4

- La “ronda” puede deducirse con sólo una parte de la clave primaria, y sin embargo la estamos repitiendo en cada set.
- La FNBC impide que esto suceda prohibiendo que existan dependencias parciales de una clave candidata, inclusive de atributos primos.
- Una dependencia que nos molesta es {nombre_torneo, año, tenista1, tenista2} → ronda, porque {nombre_torneo, año, tenista1, tenista2} no es superclave.

Forma normal Boyce-Codd (FNBC)

Ejemplo: Base de datos de torneos de tenis individual

■ Lo resolvemos de la siguiente forma:

- Torneos(nombre_torneo, ciudad, país)
- Rondas(nombre_torneo, año, tenista1, tenista2, ronda)
- Partidos(nombre_torneo, año, tenista1, tenista2, set, punt1, punt2)

TORNEOS

nombre_torneo	ciudad	país
Roland Garros	París	Francia
Masters de Madrid	Madrid	España

RONDAS

nombre_torneo	año	tenista1	tenista2	ronda
Roland Garros	2016	A. Murray	S. Wawrinka	2-final
Masters de Madrid	2015	R. Federer	R. Nadal	4-final
Roland Garros	2016	N. Djokovic	A. Murray	Final

PARTIDOS

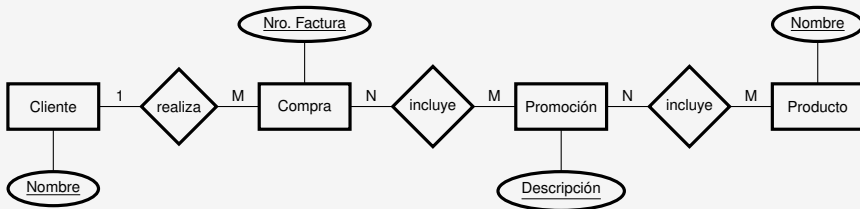
nombre_torneo	año	tenista1	tenista2	set	punt1	punt2
Roland Garros	2016	A. Murray	S. Wawrinka	1	6	4
Roland Garros	2016	A. Murray	S. Wawrinka	2	6	2
Roland Garros	2016	A. Murray	S. Wawrinka	3	4	6
Roland Garros	2016	A. Murray	S. Wawrinka	4	6	2
Masters de Madrid	2015	R. Federer	R. Nadal	1	3	6
Masters de Madrid	2015	R. Federer	R. Nadal	2	1	6
Roland Garros	2016	N. Djokovic	A. Murray	1	6	3
Roland Garros	2016	N. Djokovic	A. Murray	2	1	6
Roland Garros	2016	N. Djokovic	A. Murray	3	6	2
Roland Garros	2016	N. Djokovic	A. Murray	4	6	4

- 1 Introducción
- 2 Dependencias funcionales
- 3 Formas normales: 1FN, 2FN, 3FN, FNBC
- 4 Dependencias multivaluadas y 4FN**
- 5 Dependencias de junta y 5FN
- 6 Bibliografía

Dependencia multivaluada

Situación

- Observemos el siguiente caso de un supermercado:



Hipótesis: Por cada compra sólo se puede adquirir una vez cada promoción.

PROMOCIONES VENDIDAS

nro_factura	nombre_cliente	descripción_promo	nombre_producto
0249-19855	Juana Auzqui	Fiesta-Pancho	Pack salchichas x6
0249-19855	Juana Auzqui	Fiesta-Pancho	Pack pan de viena x6
0249-19855	Juana Auzqui	Fiesta-Pancho	Mayonesa 250gr
0034-20329	Bernardo Lühn	Vajilla Reluciente	Esponjas x2
0034-20329	Bernardo Lühn	Vajilla Reluciente	1 detergente Universo
0034-20329	Bernardo Lühn	Vajilla Reluciente	1 antigrasa Universo
0034-20329	Bernardo Lühn	Vajilla Reluciente	Repasadores x3
0058-91330	Bernardo Lühn	Fiesta-Pancho	Pack salchichas x6
0058-91330	Bernardo Lühn	Fiesta-Pancho	Pack pan de viena x6
0058-91330	Bernardo Lühn	Fiesta-Pancho	Mayonesa 250gr

Dependencia multivaluada

Situación

PROMOCIONES VENDIDAS			
nro_factura	nombre_cliente	descripción_promo	nombre_producto
0249-19855	Juana Auzqui	Fiesta-Pancho	Pack salchichas x6
0249-19855	Juana Auzqui	Fiesta-Pancho	Pack pan de viena x6
0249-19855	Juana Auzqui	Fiesta-Pancho	Mayonesa 250gr
0034-20329	Bernardo Lühn	Vajilla Reluciente	Esponjas x2
0034-20329	Bernardo Lühn	Vajilla Reluciente	1 detergente Universo
0034-20329	Bernardo Lühn	Vajilla Reluciente	1 antigraza Universo
0034-20329	Bernardo Lühn	Vajilla Reluciente	Repasadores x3
0058-91330	Bernardo Lühn	Fiesta-Pancho	Pack salchichas x6
0058-91330	Bernardo Lühn	Fiesta-Pancho	Pack pan de viena x6
0058-91330	Bernardo Lühn	Fiesta-Pancho	Mayonesa 250gr

- Clave de la relación:
 - $\{nro_factura, descripción_promo, nombre_producto\}$.
- No es cierto que “nombre_producto” dependa funcionalmente de “descripción_promo”. Sin embargo, tenemos información redundante porque *los productos que integran cada promo son siempre los mismos independientemente de quienes compran la promo.*
- Este tipo de redundancia es capturado por el concepto de **dependencia multivaluada**.

Dependencia multivaluada

Definición

- Dada una relación $R(A)$, la **dependencia multivaluada** $X \twoheadrightarrow Y$ es una restricción sobre las posibles tuplas de R que implica que para todo par de tuplas t_1, t_2 tales que $t_1[X] = t_2[X]$, deberían existir otras dos tuplas t_3 y t_4 que resulten de intercambiar los valores de Y entre t_1 y t_2 . En otras palabras, tales que:
 - $t_3[X] = t_4[X] = t_1[X] = t_2[X]$
 - $t_3[Y] = t_1[Y]$ y $t_4[Y] = t_2[Y]$
 - $t_3[A - (X \cup Y)] = t_2[A - (X \cup Y)]$ y $t_4[A - (X \cup Y)] = t_1[A - (X \cup Y)]$
- Por una cuestión de simetría, si $X \twoheadrightarrow Y$ entonces también vale que $X \twoheadrightarrow A - (X \cup Y)$.
 - Observemos que en el ejemplo anterior:
 $descripcion_promo \twoheadrightarrow nombre_producto$
 $descripcion_promo \twoheadrightarrow \{nombre_factura, nombre_cliente\}$.
- Las dependencias multivaluadas en las que $X \cup Y = A$ ó $Y \subset X$ son **triviales**.

Cuarta forma normal (4FN)

Definición

- Una relación R está en **cuarta forma normal** cuando para toda dependencia multivaluada no trivial $X \twoheadrightarrow Y$, X es superclave.
- Propiedad: Si R está en 4FN, entonces R está en FNBC.
 - ¿Demostración?
 - Toda dependencia funcional es una dependencia multivaluada:
$$X \rightarrow Y \Rightarrow X \twoheadrightarrow Y$$
 - Luego, si un esquema está en 4FN, no puede haber una df no trivial $X \rightarrow Y$ en la que X no sea superclave.
- Es común que las dependencias multivaluadas provengan de la existencia de atributos multivaluados en el modelo conceptual, o de interrelaciones N-N no capturadas.

Cuarta forma normal (4FN)

Solución al ejemplo de las promociones

- Primero normalizamos para llevar a FNBC eliminando la dependencia funcional parcial *nro_factura* \rightarrow *nombre_cliente*. Para ello descomponemos en:

Descomposición a FNBC

ClienteFactura(nro_factura, nombre_cliente)

PromoProdFactura(nro_factura, descripción_promo, nombre_producto)

- Luego eliminamos la dependencia multivaluada *descripcion_promo* \twoheadrightarrow *nombre_producto* descomponiendo en:

Descomposición a 4FN

Promociones(descripción_promo, nombre_producto)

ClientesFactura(nro_factura, nombre_cliente)

PromocionesFactura(nro_factura, descripción_promo)

Cuarta forma normal (4FN)

Solución al ejemplo de las promociones

PROMOCIONES

descripción_promo	nombre_producto
Fiesta-Pancho	Pack salchichas x6
Fiesta-Pancho	Pack pan de viena x6
Fiesta-Pancho	Mayonesa 250gr
Vajilla Reluciente	Esponjas x2
Vajilla Reluciente	1 detergente Universo
Vajilla Reluciente	1 antigraza Universo
Vajilla Reluciente	Repasadores x3

CLIENTESFACTURA

nro_factura	nombre_cliente
0249-19855	Juana Auzqui
0034-20329	Bernardo Lühn

PROMOCIONESFACTURA

nro_factura	descripción_promo
0249-19855	Fiesta-Pancho
0034-20329	Vajilla Reluciente
0058-91330	Fiesta-Pancho

- 1 Introducción
- 2 Dependencias funcionales
- 3 Formas normales: 1FN, 2FN, 3FN, FNBC
- 4 Dependencias multivaluadas y 4FN
- 5 Dependencias de junta y 5FN**
- 6 Bibliografía

Quinta forma normal (5FN)

Dependencias de junta

- Resultado [Fagin, 1977]: Siempre que en una relación $R(X, Y, Z)$ haya una dependencia multivaluada $X \twoheadrightarrow Y$ (y recuerde que, en particular, si $X \rightarrow Y$ entonces también $X \twoheadrightarrow Y$), R puede ser descompuesta sin pérdida en:
 - $R_1(X, Y)$
 - $R_2(X, Z)$
- La inversa también es cierta.
- Sin embargo, existen relaciones que pueden ser descompuestas en más de dos relaciones, también sin pérdida. Cuando esto ocurre, decimos que hay una **dependencia de junta**.
- Dada una relación $R(A)$, y una serie de subconjuntos de sus atributos, X_1, X_2, \dots, X_n , con $X_i \subset A$, decimos que (X_1, X_2, \dots, X_n) es una dependencia de junta cuando la descomposición de R en $\pi_{X_1}(R), \pi_{X_2}(R), \dots, \pi_{X_n}(R)$ es sin pérdida de información. Es decir: $\pi_{X_1}(R) * \pi_{X_2}(R) * \dots * \pi_{X_n} = R$.

Quinta forma normal (5FN)

Situación

- Un supermercado tiene varias sucursales que comercializan distintos tipos de productos (p.ej., lácteos, vinos, elementos de bazar, yerbas, etc). El supermercado trabaja con distintos proveedores y no todos comercializan todo. Pero cuando una sucursal trabaja con un cierto proveedor, le adquiere todos los productos que la sucursal comercializa y el proveedor ofrece.

COMERCIALIZACIÓN

sucursal	proveedor	tipo_producto
Floresta	El Picadero	Vinos
Floresta	La Bondad	Leches
Floresta	La Bondad	Yerbas
La Boca	Blanquín	Bazar
La Boca	Blanquín	Pañales
La Boca	Pirulo	Bazar
Villa del Parque	El Picadero	Vinos
Villa del Parque	El Picadero	Quesos
Villa del Parque	Blanquín	Bazar
Recoleta	Blanquín	Bazar
Recoleta	Romualdo	Quesos
Recoleta	El Picadero	Vinos
Recoleta	El Picadero	Quesos

- Observación: Floresta no puede comprarle Quesos a Romualdo. ¿Por qué?

Quinta forma normal (5FN)

Situación

COMERCIALIZACIÓN		
sucursal	proveedor	tipo_producto
Floresta	El Picadero	Vinos
Floresta	La Bondad	Leches
Floresta	La Bondad	Yerbas
La Boca	Blanquín	Bazar
La Boca	Blanquín	Pañales
La Boca	Pirulo	Bazar
Villa del Parque	El Picadero	Vinos
Villa del Parque	El Picadero	Quesos
Villa del Parque	Blanquín	Bazar
Recoleta	Blanquín	Bazar
Recoleta	Romualdo	Quesos
Recoleta	El Picadero	Vinos
Recoleta	El Picadero	Quesos

- No podemos identificar ninguna dependencia multivaluada.
- Sin embargo la relación puede ser descompuesta en:

Descomposición

ProveedoresSucursales(sucursal, proveedor)

ProductosProveedores(proveedor, tipo_producto)

ProductosSucursales(sucursal, tipo_producto)

Quinta forma normal (5FN)

Solución al ejemplo de las sucursales

PROVEEDORES SUCURSALES

sucursal	proveedor
Floresta	El Picadero
Floresta	La Bondad
La Boca	Blanquín
La Boca	Pirulo
Villa del Parque	El Picadero
Villa del Parque	Blanquín
Recoleta	Blanquín
Recoleta	Romualdo
Recoleta	El Picadero

PRODUCTOS PROVEEDORES

proveedor	tipo_producto
El Picadero	Vinos
La Bondad	Leches
La Bondad	Yerbas
Blanquín	Bazar
Blanquín	Pañales
Pirulo	Bazar
El Picadero	Quesos
Romualdo	Quesos

PRODUCTOS SUCURSALES

sucursal	tipo_producto
Floresta	Vinos
Floresta	Leches
Floresta	Yerbas
La Boca	Bazar
La Boca	Pañales
Villa del Parque	Vinos
Villa del Parque	Quesos
Villa del Parque	Bazar
Recoleta	Quesos
Recoleta	Vinos

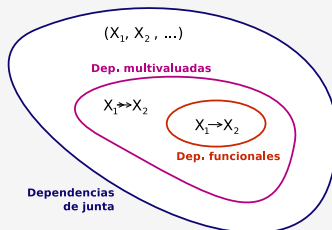
→ Existe la siguiente dependencia de junta:

$\{\{sucursal, proveedor\}, \{proveedor, tipo_producto\}, \{sucursal, tipo_producto\}\}.$

Quinta forma normal (5FN)

Definición

- Una relación $R(A)$ está en **quinta forma normal (5FN)** si y sólo si para toda dependencia de junta (X_1, X_2, \dots, X_n) *no trivial* (i.e., tal que ningún $X_i = A$) todos los X_i son superclaves.
- Observemos que en particular las dependencias funcionales y las multivaluadas son también dependencias de junta.



- Es muy difícil detectar dependencias de junta en forma general, y esta descomposición rara vez es aplicada.

- 1 Introducción
- 2 Dependencias funcionales
- 3 Formas normales: 1FN, 2FN, 3FN, FNBC
- 4 Dependencias multivaluadas y 4FN
- 5 Dependencias de junta y 5FN
- 6 Bibliografía**

Bibliografía

[ELM16] Fundamentals of Database Systems, 7th Edition.

R. Elmasri, S. Navathe, 2016.

Capítulo 14

[SILB19] Database System Concepts, 7th Edition.

A. Silberschatz, H. Korth, S. Sudarshan, 2019.

Capítulo 7

[GM09] Database Systems, The Complete Book, 2nd Edition.

H. García-Molina, J. Ullman, J. Widom, 2009.

Capítulo 3

[CONN15] Database Systems, a Practical Approach to Design, Implementation and Management, 6th Edition.

T. Connolly, C. Begg, 2015.

Capítulo 14, Capítulo 15