

# **CUADERNO DE TRABAJO:**

## **EL PROCESO DE NORMALIZACIÓN DE BASES DE DATOS**

(PROCESO DE NORMALIZACIÓN)

Cursos que apoyan:

- *Base de Datos II del Programa educativo de Ingeniería en Sistemas computacionales. (Secuencia II)*
- *Bases de datos multimedia del programa educativo de Ingeniería en diseño multimedia. (Secuencia II)*
- *Base de Datos II del Programa Educativo de Licenciatura en Tecnologías de Información. (Secuencia II)*
- *Base de Datos II del Programa Educativo de Ingeniería en Computación. (Secuencia II)*

PRESENTAN:

**Dr. José Felipe Cocón Juárez**

**Dra. Beatriz Herrera Sánchez**

Academia de Bases de datos y tratamiento de la información

APROBADO: Jueves 29 de septiembre de 2016

Ciudad del Carmen, Campeche, México.

## Contenido

Introducción.....	4
1 Proceso de Normalización hasta 3FN .....	5
Saberes a reforzar. ....	5
Estrategias metodológicas .....	5
Lecturas .....	5
Ejercicios resueltos .....	9
Primer ejemplo.....	9
Segundo ejemplo.....	12
Tercer ejemplo.....	14
Ejercicios a resolver.....	15
Ejercicio 1.1 Control de Proyectos De Investigación .....	15
Ejercicio 1.2 Control de Constancia.....	16
Ejercicio 1.3 Centro de Rehabilitación Social .....	17
Ejercicio 1.4 Relación Profesores_Proyectos .....	18
Ejercicio 1.5 Renta de Autobuses Universitarios .....	19
Ejercicio 1.6 Relación Ventas_Especies_Marina.....	20
Ejercicio 1.7 Organizaciones No Gubernamentales .....	21
Ejercicio 1.8 Control de Temas Para Tesistas .....	22
Ejercicio 1.8 Entrega de Especies Marinas.....	23
Ejercicio 1.9 Control de Cursos Online .....	24
2 Proceso de Normalización hasta 4FN .....	25
Saberes a reforzar. ....	25
Estrategias metodológicas .....	25
Lecturas .....	25
Ejemplo Resuelto 4FN explicado detalladamente .....	58
Ejercicios resueltos .....	62
Primer ejemplo.....	62
Segundo ejemplo:.....	63
Tercer ejemplo.....	63
Ejercicios a resolver.....	66

Ejercicio 2.1 Relación Inventario Cables.....	66
Ejercicio 2.2 Tienda Virtual Tennis .....	67
Ejercicio 2.3 Relación Ventas Zapaterías .....	68
Ejercicio 2.4 Relación Ventas Material Eléctrico.....	69
Ejercicio 2.5 Relación Kardex .....	70
Ejercicio 2.6 Relación Profesores Proyectos .....	71
Ejercicio 2.7 Relación Óptica .....	72
Ejercicio 2.8 Cliente óptica.....	73
Clave de respuestas.....	74
Referencias: .....	75

## Introducción

La normalización es el proceso de organizar los datos de una base de datos. Se incluye la creación de tablas y el establecimiento de relaciones entre ellas según reglas diseñadas tanto para proteger los datos como para hacer que la base de datos sea más flexible al eliminar la redundancia y las dependencias incoherentes.

Los datos redundantes desperdician el espacio de disco y crean problemas de mantenimiento. Si hay que cambiar datos que existen en más de un lugar, se deben cambiar de la misma forma exactamente en todas sus ubicaciones. Un cambio en la dirección de un cliente es mucho más fácil de implementar si los datos sólo se almacenan en la tabla Clientes y no en algún otro lugar de la base de datos.

En este sentido, se propone este cuaderno de trabajo con ejercicios prácticos de entornos reales para la aplicación del proceso de normalización utilizando las dependencias funcionales. La aplicación de dependencias funcionales apoya al proceso de normalización ya que se demuestra mediante un árbol de dependencias para la creación de las tablas.

Una vez que se tienen las tablas normalizadas se dice que se pueden crear dentro del Sistema de Gestión/Manejador de Bases de Datos para ejecutarlas sin error alguno, ya que el proceso de normalización nos ayuda a tener una redundancia controlada en ciertas tablas.

A manera general se puede decir que este cuaderno de trabajo coadyuva al curso de bases de datos II, afectando directamente a la secuencia II impactando al tema de normalización. Asimismo, impacta en el curso de bases de datos multimedia, directamente en la secuencia II sobre el tema del proceso de normalización. Las bases de datos son el andamiaje de la ingeniería básicas de la carrera.

## 1 Proceso de Normalización hasta 3FN

En este proceso de normalización nos enfocaremos hasta la tercera forma normal también conocida como 3FN, habiendo estudiado la primera forma normal y la segunda forma normal.

### Saberes a reforzar.

Se considera la teoría vistas en clases sobre las formas normales y la aplicación de dichas teorías. Así que cognitivamente abordaremos la teoría básica del proceso de normalización y las dependencias funcionales. Las habilidades psicomotoras se desarrollan mediante las prácticas o resolución del proceso de normalización.

### Estrategias metodológicas

Se propone la realización de ejercicios prácticos mediante el uso de un ordenador o computadora. Aplicando la metodología de POL (*Project Oriented Learning*) - Aprendizaje orientado a proyectos para la resolución de problemas en entornos reales.

### Lecturas

#### Objetivo de la normalización

Las bases de datos relacionales se normalizan para:

- ✓ Evitar la redundancia de los datos.
- ✓ Evitar problemas de actualización de los datos en las tablas.
- ✓ Proteger la integridad de los datos.

En el modelo relacional es frecuente llamar tabla a una relación, aunque para que una tabla bidimensional sea considerada como una relación tiene cumplir con algunas restricciones:

- ✓ Cada columna debe tener su nombre único.
- ✓ No puede haber dos filas iguales. No se permiten los duplicados.
- ✓ Todos los datos en una columna deben ser del mismo tipo.

#### Terminología equivalente

- ✓ **Relación** = Tabla o Archivo
- ✓ **Tupla** = Registro, Fila o Renglón
- ✓ **Atributo** = Campo o Columna
- ✓ **Base De Datos** = Banco de Datos
- ✓ **Dependencia Multivaluada** = Dependencia Multivalor
- ✓ **Clave** = Llave
- ✓ **Clave Primaria** = Superclase
- ✓ **Clave Ajena** = Clave Extranjera o Clave Foránea

- ✓ **Rdbms** = Del inglés *Relational Data Base Manager System*. Que Significa: Sistema Gestor de Base De Datos Relacionales

## Reglas de normalización

### 3.3.2. Reglas de normalización

La teoría de la normalización está basada en la aplicación de una serie de reglas a las que se les denomina *Reglas de normalización*. Se dice que una relación está en una determinada forma normal si satisface un cierto conjunto específico de restricciones impuestas por la regla de normalización correspondiente.

La aplicación de una regla de normalización es una operación que toma una relación como argumento de entrada y da como resultado dos o más relaciones, y:

- La relación objeto de la aplicación de la regla es desestimada en el nuevo esquema relacional considerado.
- No se introducen nuevos atributos en el esquema relacional resultante de la normalización.
- Los atributos de la relación objeto de la normalización pasan a formar parte de la intención de una o más de las relaciones resultantes.
- En la aplicación de la regla de normalización se ha debido eliminar, al menos, una dependencia existente entre los atributos de la relación objeto de la normalización.

De esta forma, la sucesiva aplicación de las reglas de normalización va a dar lugar a la generación de un número mayor de relaciones que formen parte del esquema relacional y, desde un punto de vista sólo lógico, una redundancia de los atributos considerados en el esquema.

La aplicación sucesiva de las reglas de normalización restringe, por tanto, el número de relaciones que las satisfacen. Por regla general, se dice que un esquema relacional es consistente si las relaciones satisfacen al menos la forma normal de *Boyce-Codd*.

## Proceso de normalización

### 3.3.2.1. LA PRIMERA FORMA NORMAL FN1

*Una relación R satisface la primera forma normal (FN1) si, y sólo si, todos los dominios subyacentes de la relación R contienen valores atómicos.*

La aplicación de esta regla es fácil y directa para cualquier relación, y este proceso se realiza de forma automática en el proceso de análisis del dominio del problema. Simplemente, consiste en descomponer aquellas tuplas en las que los atributos tengan más de un valor en tantas tuplas como valores estén presentes. De hecho, es una restricción innata al propio modelo relacional.

El que una relación se encuentre en *FN1* no es condición suficiente, aunque sí necesaria, para garantizar la consistencia del esquema relacional.

### 3.3.2.2. LA SEGUNDA FORMA NORMAL FN2

*Una relación R satisface la segunda forma normal (FN2) si, y sólo si, satisface la primera forma normal y cada atributo de la relación depende funcionalmente de forma completa de la clave primaria de esa relación.*

Para explicar la naturaleza de esta forma normal se va a analizar una nueva relación denominada *Matricula*, como se muestra a continuación:

---

#### Esquema-1

*Matricula (dni, asignatura#, apellido, nombre, nota, curso, aula, lugar)*

en la que:

- El atributo *asignatura#* representa la identificación de las asignaturas en las que se encuentra matriculado cada uno de los alumnos.
- El atributo *aula* representa las aulas en las que se imparte la docencia de las asignaturas.
- El atributo *lugar* representa los lugares de estudio en los que se imparte la docencia correspondiente a las asignaturas.
- El atributo *curso* representa el curso en el que se imparte la docencia de una asignatura. Por tanto, se va a suponer que existe una dependencia funcional entre los atributos *asignatura#* y *curso*, de la forma: *Matricula.asignatura# → Matricula.curso*, que representa que si bien en un curso se puede impartir docencia para varias asignaturas, una asignatura está asignada a la docencia de un único curso.

- La relación *Imparte*, se encuentra en *FN2*. La clave de esta relación es el atributo *asignatura#*, y el único atributo no primo (*curso*) depende funcionalmente de forma completa de la clave de la relación.
- Al eliminarse el atributo *curso* en la relación *Matricula-2*, se ha eliminado la dependencia funcional no completa entre el atributo *curso* y la clave de la relación (la existente entre los atributos *asignatura#* y *curso*) y, por lo tanto, los problemas que causaba esta dependencia en los procesos de mantenimiento de la información para esta relación.
- No se ha producido pérdida de información, puesto que el atributo *asignatura#* debe definirse como clave foránea de la relación *Imparte*. De esta forma, el atributo *asignatura#* en la relación *Matricula-2* para cualquier tupla, sólo podrá tomar valores existentes en alguna tupla de la relación *Imparte*. Así, basándose en esta referencia podrá conocerse en cada momento en qué curso es impartida cada asignatura para cada uno de los alumnos matriculados.

El mismo razonamiento debe hacerse para la dependencia funcional no completa existente entre la clave de la relación *Matricula-2* y el agregado (*apellidos + nombre*), puesto que este agregado sólo depende del atributo *dni* —el identificador del alumno— y no de las asignaturas —valores del atributo *asignatura#*— en las que se encuentra matriculado. Si se aplica, por tanto, a la relación *Matricula-2* la *FN2* para eliminar esta dependencia, se obtiene el siguiente esquema:

#### **Esquema-3**

<b>Imparte</b>	<i>(asignatura#, curso)</i>
<b>Alumno-2</b>	<i>(dni, apellidos, nombre)</i>
<b>Matricula-3</b>	<i>(dni, asignatura#, nota, aula, lugar)</i>

#### **3.3.2.3. LA TERCERA FORMA NORMAL *FN3***

*Una relación R satisface la tercera forma normal (FN3) si, y sólo si, satisface la segunda forma normal y cada atributo no primo de la relación no depende funcionalmente de forma transitiva de la clave primaria de esa relación. Es decir, no pueden existir dependencias entre los atributos que no forman parte de la clave primaria de la relación R.*

Se puede observar que la relación *Matricula-3*, la cual se encuentra en *FN2*, sigue presentando problemas en los procesos de manipulación de la misma. Los problemas son debidos a que existe una dependencia entre los atributos *aula* y *lugar*.

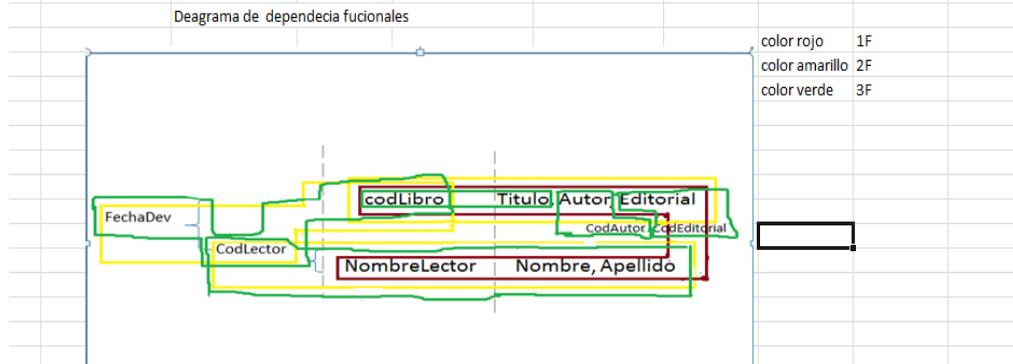
Si se considera, como es lógico, que cada aula se encuentra ubicada físicamente en un único lugar, y que la docencia de una determinada asignatura —para un determinado conjunto de alumnos; es decir, diferentes alumnos matriculados en una asignatura pueden recibir docencia en aulas diferentes (grupos de clase)— se imparte en una única aula, se observa que existe una dependencia funcional entre los atributos no primos *lugar* y *aula*, además de las dependencias funcionales completas entre los atributos *lugar* y *aula* con la clave de la relación *Matricula-3*.

## Ejercicios resueltos

### Ejemplos de normalización:

**Primer ejemplo.** Aplicamos la normalización en el siguiente problema planteado de una biblioteca la cual nos proporciona los siguientes datos:

CodLibro	Titulo	Autor	Editorial	NombreLector	FechaDev
1001	Variable compleja	Murray Spiegel	McGraw Hill	Pérez Gómez, Juan	15/04/2005
1004	Visual Basic 5	E. Petroussos	Anaya	Ríos Terán, Ana	17/04/2005
1005	Estadística	Murray Spiegel	McGraw Hill	Roca, René	16/04/2005
1006	Oracle University	Nancy Greenb	Oracle Corp.	García Roque, Luis	20/04/2005
1007	Clipper 5.01	Ramalho	McGraw Hill	Pérez Gómez, Juan	18/04/2005



## Primera Forma normal

CodLibro	Titulo	Autor	Editorial	Paterno	Materno	Nombres	FechaDev
1001	Variable compleja	Murray Spiegel	McGrawHill	Pérez	Gómez	Juan	15/04/2005
1004	Visual Basic 5	E. Petroustos	Anaya	Ríos	Terán	Ana	17/04/2005
1005	Estadística	Murray Spiegel	McGrawHill	Roca		René	16/04/2005
1006	OracleUniversity	NancyGreenberg	OracleCorp.	García	Roque	Luis	20/04/2005
1006	OracleUniversity	Priya Nathan	OracleCorp.	García	Roque	Luis	20/04/2005
1007	Clipper 5.01	Ramalho	McGrawHill	Pérez	Gómez	Juan	18/04/2005

## Segunda Forma Normal

Tabla libro				Tabla lector			
CodLibro	Titulo	Autor	Editorial	CodLector	Paterno	Materno	Nombres
1001	Variable comp	Murray Spiegel	McGrawHill	501	Pérez	Gómez	Juan
1004	Visual Basic 5	E. Petroustos	Anaya	502	Ríos	Terán	Ana
1005	Estadistica	Murray Spiegel	McGrawHill	503	Roca		René
1006	Oracle Univers	NancyGreenberg	Oracle Corp.	504	Garcia	Roque	Luis
1006	Oracle Univers	Priya Nathan	Oracle Corp.				
1007	Clipper 5.01	Ramalho	McGrawHill				

Tabla prestamo		
CodLibro	CodLector	FechaDev
1001	501	15/04/2005
1004	502	17/04/2005
1005	503	16/04/2005
1006	504	20/04/2005
1007	501	18/04/2005

## Tercera Forma Normal

Tabla libro		Tabla autor		Tabla edición		Tabla entrega		
CodLibro	Título	CodAutor	Autor	CodEditorial	Editorial	CodLibro	CodLector	FechaDev
1001	Variable compleja	801	Murray Spiegel	901	McGraw Hill	1001	501	15/04/2005
1004	Visual Basic 5	802	E. Petroustos	902	Anaya	1004	502	17/04/2005
1005	Estadística	803	Nancy Greenberg	903	Oracle Corp.	1005	503	16/04/2005
1006	Oracle University	804	Priya Nathan			1006	504	20/04/2005
1007	Clipper 5.01	806	Ramalho			1007	501	18/04/2005

Tabla codigo		Tabla registro		Tabla lector			
CodLibro	codAutor	CodLibro	codEditorial	CodLector	Paterno	Materno	Nombres
1001	801	1001	901	501	Pérez	Gómez	Juan
1004	802	1004	902	502	Ríos	Terán	Ana
1005	801	1005	901	503	Roca		René
1006	803	1006	903	504	García	Roque	Luis
1006	804	1007	901				
1007	806						

## Segundo ejemplo.

En este ejercicio se aplica la Normalización correspondiente con los datos proporcionados por el usuario sobre un control de proyecto

	A	B	C	D	E	F	G	H	I	J	K
1	CONTROL DE PROYECTOS DE INVESTIGACION										
2	NUMPROY	NOMPROY	NUMPART	NOMPART	APEPART	RESPONSABLEPAR	MONTOPROY	CODLINEA	NOMLINEA	COD_CULTIVA	CULTIVA_TEMAS
3	1111 MADUREZ	1111 FELIPE	COCON	-		500000	100 E-LEARNING			1 OBJETO APRENDIZAJE	
4	1111 MADUREZ	1112 PATRICIA	ZAVAleta		1111	500000	100 E-LEARNING			2 TUTORIA ONLINE	
5	1111 MADUREZ	1113 JORGE	VANOYE		1111	500000	100 E-LEARNING			3 BIOIFNORMATICA	
6	1111 MADUREZ	1114 DAMARIS	PEREZ		1111	500000	100 E-LEARNING			2 TUTORIA ONLINE	
7	1111 MADUREZ	1115 VERONICA	VELUETA		1111	500000	100 E-LEARNING			5 RISK IT	
8	1112 TUTORIA	1111 FELIPE	COCON		1112	500000	200 GESTION			6 ROA	
9	1112 TUTORIA	1112 PATRICIA	ZAVAleta	-		500000	200 GESTION			2 TUTORIA ONLINE	
10	1112 TUTORIA	1113 JORGE	VANOYE		1112	500000	200 GESTION			3 BIOIFNORMATICA	
11	1112 TUTORIA	1114 DAMARIS	PEREZ		1112	500000	200 GESTION			2 TUTORIA ONLINE	
12	1112 TUTORIA	1115 VERONICA	VELUETA		1112	500000	200 GESTION			7 VAL IT	
13											

## Primera Forma Normal

Portapapeles	Fuente	Alineación	Número	Estilos	Celdas	Modificar
M17	f <sub>x</sub>					
A	B	C	D	E	F	G
1						
2	PARTICIPANTES_LINEAS					
3	NUMPART	NOMPART	APEPART	CODLINEA	NOMLINEA	COD_CULTIVA CULTIVA_TEMAS
4	1111	FELIPE	COCON	100	E-LEARNING	1 OBJETO APRENDIZAJE
5	1112	PATRICIA	ZAVAleta	100	E-LEARNING	2 TUTORIA ONLINE
6	1113	JORGE	VANOYE	100	E-LEARNING	3 BIOIFNORMATICA
7	1114	DAMARIS	PEREZ	100	E-LEARNING	2 TUTORIA ONLINE
8	1115	VERONICA	VELUETA	100	E-LEARNING	5 RISK IT
9	1111	FELIPE	COCON	200	GESTION	6 ROA
10	1112	PATRICIA	ZAVAleta	200	GESTION	2 TUTORIA ONLINE
11	1113	JORGE	VANOYE	200	GESTION	3 BIOIFNORMATICA
12	1114	DAMARIS	PEREZ	200	GESTION	2 TUTORIA ONLINE
13	1115	VERONICA	VELUETA	200	GESTION	7 VAL IT
14						
15						
16						
17	PROYECTOS					
18	NUMPROY	NOMPROY				
19	1111	MADUREZ				
20	1112	TUTORIA				
21						

## Segunda Forma Normal

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2		PROYECTOS										
3		NUMPROY	NOMPROY									
4		1111	MADUREZ									
5		1112	TUTORIA									
6												
7												
8		LINEAS										
9		CODLINEA	NOMLINEA									
10		100	E-LEARNING									
11		200	GESTION									
12												
13												
14												
15												
16												
17												
18												
19												
20												
21												
22												
23												

## Tercera Forma Normal

	A	B	C	D	E	F	G	H	I	J	K	
1	PROYECTOS			PARTICIPANTES								
2	NUMPROY	NOMPROY		NUMPART	NOMPART	APEPART						
3	1111	MADUREZ		1111	FELIPE	COCON						
4	1112	TUTORIA		1112	PATRICIA	ZAVALET						
5				1113	JORGE	VANOYE						
6				1114	DAMARIS	PEREZ						
7				1115	VERONICA	VELUETA						
8	LINEAS											
9	CODLINEA	NOMLINEA										
10	100	E-LEARNING										
11	200	GESTION										
12												
13												
14	PARTICIPANTE LINEA			CULTIVA LINEAS								
15	NUMPART	CODLINEA		CODLINEA	COD CULTIVA							
16	1111	100		100	1							
17	1112	100		100	2							
18	1113	100		100	3							
19	1114	100		100	5							
20	1115	100		200	1							
21	1111	200		200	2							
22	1112	200		200	3							
23	1113	200		200	5							
24	1114	200										
25	1115	200										

### Tercer ejemplo.

#### 2.2.1.1 Primera forma normal

Nro_GI	fecha	nom_repcionista	cod_art	cant_art	nom_art
GI/000010	01/06/2010	Marcos Gonzales	AR410	12	pantalón
GI/000011	02/06/2010	José Julon	AR411	36	polos
GI/000012	03/06/2010	Samuel Jiménez	AR412	24	Medias
GI/000013	04/06/2010	Manuel Pérez	AR413	36	polos

#### 2.2.1.2 Segunda forma normal

Guia_remision			Articulo		
Nro_GI	fecha	nom_repcionista	cod_art	cant_art	nom_art
GI/000010	01/06/2010	Marcos Gonzales	AR410	12	pantalón
GI/000011	02/06/2010	José Julon	AR411	36	polos
GI/000012	03/06/2010	Samuel Jiménez	AR412	24	Medias
GI/000013	04/06/2010	Manuel Pérez	AR413	36	polos

#### 2.2.1.3 Tercera forma normal

Guia_interna		Repcionista		
Nro_GI	fecha	cod_repcionista	nom_repcionista	cant_art
GI/000010	01/06/2010	R0001	Marcos Gonzales	12
GI/000011	02/06/2010	R0002	José Julon	36
GI/000012	03/06/2010	R0003	Samuel Jiménez	24
GI/000013	04/06/2010	R0004	Manuel Pérez	36

Articulo	
cod_art	nom_art
AR410	pantalón
AR411	polos
AR412	Medias
AR413	polos

## Ejercicios a resolver

### Ejercicio 1.1 Control de Proyectos De Investigación

CONTROL DE PROYECTOS DE INVESTIGACION											
NUMPROY	NOMPROY	NUMPART	NOMPART	APEPART	RESPONSABLEPAR	MONTOPROY	CODLINEA	NOMLINEA	COD_CULTIVA	CULTIVA_TEMAS	
1111	MADUREZ	1111	FELIPE	COCON	-	500000	100	E-LEARNING	1	OBJETO APRENDIZAJE	
1111	MADUREZ	1112	PATRICIA	ZAVAleta	1111	500000	100	E-LEARNING	2	TUTORIA ONLINE	
1111	MADUREZ	1113	JORGE	VANOYE	1111	500000	100	E-LEARNING	3	BIOIFNORMATICA	
1111	MADUREZ	1114	DAMARIS	PEREZ	1111	500000	100	E-LEARNING	2	TUTORIA ONLINE	
1111	MADUREZ	1115	VERONICA	VELUETA	1111	500000	100	E-LEARNING	5	RISK IT	
1112	TUTORIA	1111	FELIPE	COCON	1112	500000	200	GESTION	6	ROA	
1112	TUTORIA	1112	PATRICIA	ZAVAleta	-	500000	200	GESTION	2	TUTORIA ONLINE	
1112	TUTORIA	1113	JORGE	VANOYE	1112	500000	200	GESTION	3	BIOIFNORMATICA	
1112	TUTORIA	1114	DAMARIS	PEREZ	1112	500000	200	GESTION	2	TUTORIA ONLINE	
1112	TUTORIA	1115	VERONICA	VELUETA	1112	500000	200	GESTION	7	VALIT	

## Ejercicio 1.2 Control de Constancia

CONTROL DE CONSTANCIAS									
NO.EMP	NOMBRE_E	APELLIDO_E	NO.CERTIFICA	CVE_TIPO	TIPO_CERTIFICA	F_EXPEDICION	MAT_ESTU	NOM_ESTU	CARRERA
1567	NELLY	ANGEL	513	1	PARCIAL	12-jul-12	901144	JUAN	ICO
1567	NELLY	ANGEL	514	2	COMPLETO	13-jul-12	901144	JUAN	ICO
1567	NELLY	ANGEL	515	1	PARCIAL	15-jul-13	902245	LUIS	ISC
1567	NELLY	ANGEL	516	2	COMPLETO	18-sep-13	902245	LUIS	ISC
1567	NELLY	ANGEL	517	3	PROMEDIO/SEM	18-sep-13	691124	MARITZA	ISC
1567	NELLY	ANGEL	518	3	PROMEDIO/SEM	18-sep-13	789066	GABRIELA	ICO
1789	DIANA	LOPEZ	519	3	PROMEDIO/SEM	18-sep-13	103066	YARITZA	ICO
1789	DIANA	LOPEZ	520	3	PROMEDIO/SEM	20-sep-13	112345	JUAN	ICO
1789	DIANA	LOPEZ	521	3	PROMEDIO/SEM	20-sep-13	114523	PEDRO	ISC
1789	DIANA	LOPEZ	522	3	PROMEDIO/SEM	20-sep-13	662345	YAIR	ICO
1789	DIANA	LOPEZ	523	3	PROMEDIO/SEM	20-sep-13	565656	JOSE	ICO
1890	ERIKA	CONTRERAS	567	2	COMPLETO	20-sep-13	556677	JOSE	ISC
1890	ERIKA	CONTRERAS	568	2	COMPLETO	20-sep-13	334455	ADRIANA	ICO
1550	GABRIELA	OROZCO	569	2	COMPLETO	20-sep-13	223344	RUBI	ISC

## Ejercicio 1.3 Centro de Rehabilitación Social

CENTRO DE REHABILITACIÓN SOCIAL																
CONTROL DE INTERNOS																
NUMREO	NOMBRE	APELLIDO	ALIAS	NUMDELITO	NOMDELITO	MIN_DELITO	MAX_DELITO	REOANNO	CODZONA	ZONA	NUMPOLICIA	NOMPOLICIA	JEFEPOLICIA	NUMCELDA	CVE_TRABajo	TRABAJO
1111	ALEXIS	CUEVAS	EL ZORRO	2501	SECUESTRO	3	5	5	4444	SECUESTROS	1001	JUAN LUIS	-	211	300	CARPINTERO
1112	JUAN	PEREZ	EL JUAN	2502	SECUESTRO EX	1	3	3	4444	SECUESTROS	1002	LUIS MIGUEL	1001	212	300	CARPINTERO
1113	JULIAN	LOPEZ	EL JULI	2601	ROBO A TRANS	1	2	1	5555	ROBOS	1003	JULIO IGLESI	1001	215	400	FONTANERO
1115	ALEX	HERNANDEZ	EL MATADOR	2602	ROBO A CASA/	1	3	3	5555	ROBOS	1004	ANDRES GAR	1002	216	400	FONTANERO
1118	JULIO	ANTONIO	LA PANZA	2603	ROBO C/VIOLE	1	5	2	5555	ROBOS	1005	JUAN ANTON	1002	217	400	FONTANERO
1119	CARLOS	CHAN	EL CRACK	2701	HOMICIDIO C/	10	15	15	6666	HOMICIDIOS	1006	ANTONIO LU	1002	300	400	FONTANERO
1110	HECTOR	GARZA	EL CHINO	2702	HOMICIDIO IN	12	18	14	6666	HOMICIDIOS	1007	FRANK QUE	1001	311	500	BILIOTECOLOG
1119	CARLOS	CHAN	EL CRACK	2703	HOMICIDIO CI	0.6	5	1	6666	HOMICIDIOS	1008	GREG LEE	1002	312	500	BILIOTECOLOG
1110	HECTOR	GARZA	EL CHINO	2704	HOMICIDIO PI	6	9	9	6666	HOMICIDIOS	1009	GREG KIRILLI	1001	315	500	BILIOTECOLOG
1112	JUAN	PEREZ	EL JUAN	2501	SECUESTRO	3	5	5	4444	SECUESTROS	1001	JUAN LUIS	-	212	300	CARPINTERO
1111	ALEXIS	CUEVAS	EL ZORRO	2502	SECUESTRO EX	1	3	3	4444	SECUESTROS	1001	JUAN LUIS	-	211	300	CARPINTERO

## Ejercicio 1.4 Relación Profesores\_Proyectos

Relación PROFESORES_PROYECTOS											
No.Emp	NomEmpleado	ApEmpleado	Cve_proy1	NomProyecto1	MontoProyecto1	Cve_proy2	NomProyecto2	MontoProyecto2	Num_Informe	Nom_Informe	...
1234	Patricia	Zavaleta	513	ROA	950,000	200	BECA	40,000	1	Reporte Parcial	...
1255	Gustavo	Verduzco	514	Redes	999,000	100	BECA	30,000	1	Reporte Parcial	...
1588	Victor	Hernández	516	Android	590,000	100	BECA	30,000	1	Reporte Parcial	...
1688	Eduardo	Orbe	517	BlackBerry	590,000	100	BECA	30,000	1	Reporte Parcial	...
1455	Eduardo	Torres	518	iOS	590,000	100	BECA	30,000	1	Reporte Parcial	...
Num_informe	Nom_informe	Num_informe	Nom_informe	Num_informe	Nom_informe	MatEstudiante	NomEstudiante	MatEstudiante2	NomEstudiante	MatEstudiante3	NomEstudiante
2	Reporte Parcial	3	Reporte Parcial	4	Reporte Final	3312	Juan Antonio	3315	Rubi del Carm	3355	Luis Miguel
2	Reporte Parcial	3	Reporte Parcial	4	Reporte Final	5512	Wilson	2255	Gabriela	6677	Cruz
2	Reporte Parcial	3	Reporte Parcial	4	Reporte Final	8812	Irvin	1233	Carlos	8811	Esmeralda
2	Reporte Parcial	3	Reporte Parcial	4	Reporte Final	9988	Enri	1243	Zafiro	8814	Diamantina
2	Reporte Parcial	3	Reporte Parcial	4	Reporte Final	6678	Raquel	1278	Rubi	8815	Jade

## Ejercicio 1.5 Renta de Autobuses Universitarios

RENTA DE AUTOBUSES UNIVERSITARIOS														
NO_AUTOBUS	MARCA	TIPO	PLAZAS	F_PRESTAMO	F_INICIO	HORA_INICIO	F_FIN	HORA_FIN	ORIGEN	DESTINO	CVE_PERSONA	NOM_PERSONA	APE_PERSONA	FACULTAD
1	MERCEDES	PASAJEROS	41	10-dic-12	15-dic-12	06:00	10-ene-13	16:00	CARMEN	CANCUN	22	JULIAN	MONTERO	NATURALES
2	TOYOTA	PASAJEROS	15	10-may-12	01-jun-12	05:00	05-jun-12	12:00	CARMEN	MERIDA	23	PEDRO	PICENT	MATEMATICAS
3	MERCEDES	PASAJEROS	35	15-abr-12	10-may-12	05:00	10-may-12	19:00	CARMEN	SABANCUY	28	ANDRES	SALAZAR	INFORMATICA

## Ejercicio 1.6 Relación Ventas\_Especies\_Marina

RELACION VENTAS_ESPECIES_MARINAS												
CVE_ESPECIES	NOM_ESPECIES	TIPO_ESPECIE	P.U.	COD_VTA	FECHA_VENTA	H_VENTA	KILOS_POR_VTA	RFC_CLIENTE	NOM_CLIENTE	APE_CLIENTE	MOSTRADOR	NOM_MOSTRAD
100	ESCAMAS	PARGO	\$ 85.00	1122	01/11/2014	14:00	2.00	BARO60	ROSARIO	BAÑOS	1	GABRIEL
100	ESCAMAS	CURVINA	\$ 50.00	1122	01/11/2014	14:00	2.00	BARO60	ROSARIO	BAÑOS	1	GABRIEL
100	ESCAMAS	ESMEDREGAL	\$ 60.00	1124	01/11/2014	15:00	5.00	LOUL79	ULISES	LOPES	1	GABRIEL
100	ESCAMAS	CURVINA	\$ 50.00	1124	02/11/2014	15:00	5.00	LOUL79	ULISES	LOPES	2	MANUEL
300	MOLUSCOS	PULPO	\$ 50.00	2233	15/10/2014	14:00	3.00	PERA80	RAMON	PEREZ	2	MANUEL
100	ESCAMAS	HUACHINANGO	\$ 90.00	2323	03/11/2014	14:15	5.00	ANMI90	MIGUEL	ANTONIO	1	GABRIEL
100	ESCAMAS	RUBIA	\$ 50.00	2424	01/11/2014	14:30	4.00	POLU86	LUIS	POOR	1	GABRIEL
200	S/ESCAMAS	TIBURON	\$ 70.00	2526	03/11/2014	14:45	6.00	PEJU89	JUNIOR	PEX	1	GABRIEL
200	S/ESCAMAS	MANTARAYA	\$ 80.00	2825	02/11/2014	10:00	3.00	BAAL90	ALEX	BANDERA	2	MANUEL
300	MOLUSCOS	PULPO	\$ 50.00	3030	15/10/2014	11:00	2.00	ANAN70	ANTONIO	ANTONIO	2	MANUEL

## Ejercicio 1.7 Organizaciones No Gubernamentales

NO. PARTICIPANTE	NOM. PARTICIPANTE	PROVINCIA	DIRECCION	TIPO PARTICPANTE	TIPO TRABAJADOR	FECHA INICIO	P. SEGURO SOCIAL	P. I.S.R.	F. NACIMIENTO	PROFESION		
1	MIGUEL	CARMEN	CALLE 24, 55	TRABAJADOR	VOLUNTARIO	10-ene-12	-	-	12-dic-65	TECNICO	...	
2	LUIS	CARMEN	CALLE 24, 65	TRABAJADOR	VOLUNTARIO	10-ene-12	-	-	10-ene-68	TECNICO	...	
3	PEDRO	CAMPECHE	CALLE 34, 680	TRABAJADOR	ASALARIADO	15-dic-12	285.75	155.5	-	-	...	
4	ANTONIO	CAMPECHE	CALLE 44, 650	TRABAJADOR	ASALARIADO	10-dic-12	285.75	155.5	-	-	..	
5	CINTHIA	CAMPECHE	CALLE 54, 165	TRABAJADOR	VOLUNTARIO	09-dic-12	-	-	15-may-76	ESTUDIANTE	...	
6	IRVIN	CARMEN	CALLE 56, 67	SOCIO	-	-	-	-	-	-	...	
7	KENIA	CARMEN	CALLE 25, 30	SOCIO	-	-	-	-	-	-	...	
8	MARITZA	CARMEN	CALLE 25, 32	SOCIO	-	-	-	-	-	-	...	
HORAS	F. ALTA SOCIO	CUOTA	APORTACION	RFC	NOM. ASOCIACION	DENOMINACION	PROVINCIA	TIPO	ID. PROYECTO	OBJETIVO	BENEFICIARIO	ZONA
4	-	-	-	CANF901203	FUNDACION CANELI DEPORTISTAS	CARMEN	INTEGRACION	1	INTEGRAR A JC JUVENTUD	SABANCUY		
4	-	-	-	CANF901203	FUNDACION CANELI DEPORTISTAS	CARMEN	INTEGRACION	1	INTEGRAR A JC JUVENTUD	SABANCUY		
	-	-	-	CANF901203	FUNDACION CANELI DEPORTISTAS	CARMEN	INTEGRACION	1	INTEGRAR A JC JUVENTUD	SABANCUY		
	-	-	-	CANF901203	FUNDACION CANELI DEPORTISTAS	CARMEN	INTEGRACION	1	INTEGRAR A JC JUVENTUD	SABANCUY		
4	-	-	-	CANF901203	FUNDACION CANELI DEPORTISTAS	CARMEN	INTEGRACION	1	INTEGRAR A JC JUVENTUD	SABANCUY		
-	05-ene-12	10,000	20,000	CANF901203	FUNDACION CANELI DEPORTISTAS	CARMEN	INTEGRACION	1	INTEGRAR A JC JUVENTUD	SABANCUY		
-	05-ene-12	10,000	25,000	CANF901203	FUNDACION CANELI DEPORTISTAS	CARMEN	INTEGRACION	1	INTEGRAR A JC JUVENTUD	SABANCUY		
-	05-ene-12	10,000	25,000	CANF901203	FUNDACION CANELI DEPORTISTAS	CARMEN	INTEGRACION	1	INTEGRAR A JC JUVENTUD	SABANCUY		

## Ejercicio 1.8 Control de Temas Para Tesistas

CONTROL DE TEMAS PARA TESISTAS										
NO.DIRECTOR	NOM_DIRECTOR	APE_DIRECTOR	NO.TEMA	NOM_TEMA	MAT_ESTUDIANTE	NOM_ESTUDIANTE	APE_ESTUDIANTE	FECHA_REGISTRO	CVE TIPO	TIPO
1 GUSTAVO	VERDUZCO		1 DOMÓTICA		112233 EMETERIO	LOPEZ		10-jul-12	1 CONCLUIDA	...
2 VICTOR	HERNANDEZ		1 DOMÓTICA		112255 JUAN	PEREZ		10-jul-12	1 CONCLUIDA	...
3 EDUARDO	ORBE		2 GRAFICO NO LINEAL		667788 ARGENTINA	SANCHEZ		14-oct-13	2 EN PROCESO	...
4 REYNA	SANCHEZ		3 VISIÓN ARTIFICIAL		334455 JUAN CARLOS	SANCHEZ		14-oct-13	2 EN PROCESO	...
FECHA_PROTO	CVE_REVISOR1	NOM_REVISOR	APE_REVISOR2	CVE_REVISOR1	NOM_REVISOR1	APE_REVISOR2	CVE_CA	NOMBRE_CA	CVE_L	NOM_LINEA
10-dic-12	11	EDUARDO	ORBE		-	-		11 CIENCIAS DE LA	1	SISTEMAS DISTRIBUIDOS
10-dic-12	11	EDUARDO	ORBE		-	-		11 CIENCIAS DE LA	1	SISTEMAS DISTRIBUIDOS
14-abr-14	22	GUSTAVO	VERDUZCO		33 VICTOR	HERNANDEZ		11 CIENCIAS DE LA	2	APLICACIONES IA
14-abr-14	11	EDUARDO	ORBE		44 ROGER	PEREZ		11 CIENCIAS DE LA	2	APLICACIONES IA

## Ejercicio 1.8 Entrega de Especies Marinas

ENTREGA DE ESPECIES MARINAS																
CVE_ESPECIES	NOM_ESPECIES	ID_TIPO_ESPECIE	TIPO_ESPECIE	F_ENTRADA	H_ENTREGA	IFE_RESP	NOMBRE	APELLIDO	N_LANCHA	TIPO_LANCHA	KG_ENTREGADOS	LTROS_GAS_CONSUMO	IFE_AYUDANTE	NOM_AYUDANTE		
100	ESCAMAS		1 PARGO	01/11/2014	14:00	112211	JUANITO	GARCIA	15	RIBERENA	10	10	884545	GUSTAVO		
100	ESCAMAS		2 ESMEDREGAL	01/11/2014	14:00	112211	JUANITO	GARCIA	15	RIBERENA	15	15	884545	GUSTAVO		
100	ESCAMAS		3 CURVINA	02/11/2014	14:00	223344	PEDRITO	LOPEZ	25	RIBERENA	25	15	995050	LUIS		
100	ESCAMAS		4 HUACHINANGO	03/11/2014	14:00	113311	LUSITO	REYES	12	C/ESTACION RIB	50	40	987070	ANDREY		
100	ESCAMAS		5 RUBIA	01/11/2014	14:00	114411	ANTONIO	PEREZ	23	C/ESTACION RIB	70	30	607070	MANUEL		
200	S/ESCAMAS		1 TIBURON	03/11/2014	14:00	113311	LUSITO	REYES	12	C/ESTACION RIB	60	40	987070	ANDREY		
200	S/ESCAMAS		1 MANTARAYA	02/11/2014	14:00	114411	ANTONIO	PEREZ	23	C/ESTACION RIB	50	20	708080	RICO		
300	MOLUSCOS		1 PULPO	15/10/2014	14:00	115511	PEPE	JESUS	28	C/ESTACION RIB	80	30	807989	ADRIAN		
300	MOLUSCOS		2 CALAMAR	15/10/2014	14:00	115511	PEPE	JESUS	28	C/ESTACION RIB	90	30	807989	ADRIAN		

## Ejercicio 1.9 Control de Cursos Online

CONTROL CURSOS ONLINE									
USUARIO	ID_CURSOS	NOM_CURSOS	CURSO_PREFERENCIA	COSTO	INICIA	TERMINA	INSCRITO_CURSO	ID_AREA	NOM_AREA
JUAN@HOTMAIL.COM	110	BIG DATA BASIC	TRUE	1200	MAYO	MAYO	110	1	INGENIERÍA Y TECNOLOGÍA
JUAN@HOTMAIL.COM	115	BIG DATA EXTENDER	FALSE	1500	JUNIO	JUNIO	-	1	INGENIERÍA Y TECNOLOGÍA
JUAN@HOTMAIL.COM	200	ESTADISTICAS PARA DUMMIES	TRUE	1000	MAYO	MAYO	200	1	INGENIERÍA Y TECNOLOGÍA
JUAN@HOTMAIL.COM	202	ESTADISTICAS PROFESIONAL	FALSE	1200	JUNIO	JUNIO	-	1	INGENIERÍA Y TECNOLOGÍA
JUAN@HOTMAIL.COM	204	ESTADISTICAS PEDAGICA	FALSE	1250	JULIO	JULIO	204	1	INGENIERÍA Y TECNOLOGÍA
PEDRO@GMAIL.COM	110	BIG DATA BASIC	TRUE	1200	MAYO	MAYO	-	1	INGENIERÍA Y TECNOLOGÍA
PEDRO@GMAIL.COM	115	BIG DATA EXTENDER	TRUE	1500	JUNIO	JUNIO	-	1	INGENIERÍA Y TECNOLOGÍA
PEDRO@GMAIL.COM	200	ESTADISTICAS PARA DUMMIES	FALSE	1000	MAYO	MAYO	-	1	INGENIERÍA Y TECNOLOGÍA
PEDRO@GMAIL.COM	202	ESTADISTICAS PROFESIONAL	FALSE	1200	JUNIO	JUNIO	202	1	INGENIERÍA Y TECNOLOGÍA
PEDRO@GMAIL.COM	204	ESTADISTICAS PEDAGICA	TRUE	1250	JULIO	JULIO	-	1	INGENIERÍA Y TECNOLOGÍA
LUIS@YAHOO.COM.MX	110	BIG DATA BASIC	FALSE	1200	MAYO	MAYO	110	1	INGENIERÍA Y TECNOLOGÍA
LUIS@YAHOO.COM.MX	115	BIG DATA EXTENDER	FALSE	1500	JUNIO	JUNIO	115	1	INGENIERÍA Y TECNOLOGÍA
LUIS@YAHOO.COM.MX	200	ESTADISTICAS PARA DUMMIES	TRUE	1000	MAYO	MAYO	-	1	INGENIERÍA Y TECNOLOGÍA
LUIS@YAHOO.COM.MX	202	ESTADISTICAS PROFESIONAL	TRUE	1200	JUNIO	JUNIO	-	1	INGENIERÍA Y TECNOLOGÍA
LUIS@YAHOO.COM.MX	204	ESTADISTICAS PEDAGICA	FALSE	1250	JULIO	JULIO	-	1	INGENIERÍA Y TECNOLOGÍA
LUIS@YAHOO.COM.MX	300	REDES PARA DUMMIES	TRUE	1000	JULIO	JULIO	-	1	INGENIERÍA Y TECNOLOGÍA

## **2 Proceso de Normalización hasta 4FN**

En este proceso de normalización nos enfocaremos nuevamente desde la primera hasta la Cuarta forma normal también conocida como 4FN, habiendo estudiado con perspectiva de otro autor, la primera forma normal, la segunda forma normal y la tercera forma normal.

### **Saberes a reforzar.**

Se considera la teoría vistas en clases sobre las formas normales y la aplicación de dichas teorías. Así que cognitivamente abordaremos la teoría básica de normalización y dependencias funcionales. Las habilidades psicomotoras se desarrollan mediante las prácticas o resolución del proceso de normalización.

### **Estrategias metodológicas**

Se propone la realización de ejercicios prácticos mediante el uso de un ordenador o computadora. Aplicando la metodología de POL (*Project Oriented Learning*)- Aprendizaje orientado a proyectos en situaciones reales de la vida cotidiana.

### **Lecturas**

Perspectiva del proceso de normalización del autor Alice Y. H. Tsai en el libro de Sistemas de bases de datos. Administración y Uso.

# 10

## Diseño de bases de datos: formas normales de relaciones

### 10.1 INTRODUCCION: EL ENFOQUE DEL DISEÑO

En la parte III se estudiaron detalladamente los enfoques arquitectónicos, la definición de datos y los lenguajes de manejo de datos de los sistemas CODASYL así como los DBMS jerárquicos y relacionales. Sin embargo, aún no se ha expuesto un problema fundamental, la manera de llegar al modelo de base de datos conveniente. Los capítulos 10 y 11 se ocupan de ello con la descripción de algunas técnicas para que, a partir de éstas, se derive el modelo conceptual a implantar mediante el DBMS.

Como se expuso en los capítulos anteriores, el esquema conceptual incluye no sólo los registros de tipo conceptual y sus campos asociados, sino también las relaciones entre los registros del mismo tipo. En un programa en lenguaje de alto nivel la definición de los datos de un esquema difiere de la definición de los datos en la inclusión de las relaciones entre los datos. El propósito de describir relaciones de datos en el esquema es el de establecer enlaces o ligas entre los tipos de registro que tienen relación entre sí para el procesamiento simultáneo de distintos archivos. En un esquema CODASYL una relación entre datos se describe con un "conjunto" con un campo de conexión (clave externa) en el modelo relacional o con la relación padre-hijo en la descripción física de la base de datos jerárquica. Así, el proceso de diseño involucra las tres fases siguientes:

- i) agrupamiento de los datos globales de una base de datos en uno o más registros de tipo conceptual (entidades)

- ii) definición de las relaciones conectando entidades relacionadas en un modelo conceptual de base de datos
- iii) transformación del modelo conceptual de la base de datos en un esquema conceptual que se pueda describir mediante un DBMS específico para la implantación de la base de datos.

De las fases i) y ii) se deriva un modelo lógico o conceptual de la base, independientemente del equipo y el DBMS a usar en la implantación. Sin embargo, las restricciones que dependen del sistema se deben tomar en cuenta en la fase iii) cuando el modelo se va a mapear en un esquema conceptual. En este capítulo se describen los métodos para obtener los registros de tipo conceptual para el modelo de la base de datos, mientras que la construcción del modelo de la base de datos y el mapeo de los esquemas conceptuales para los tipos específicos de DBMS se estudiarán en el capítulo 11.

**EL ENFOQUE INTUITIVO.** El diseño de la base de datos es una de las responsabilidades principales del DBA. Es posible mediante el sentido común llegar a modelos sencillos. Sin embargo, los modelos complejos requieren del análisis cuidadoso de las consultas usuales, los datos por usar, y así sucesivamente. Se empezará la exposición del diseño de las bases de datos con un ejemplo sencillo que da lugar al modelo de la base por medio del **enfoque intuitivo** que dará al lector alguna idea de lo que es el diseño de una base de datos. Después se presentará una técnica para modelar bases de datos llamada **proceso de normalización** que permita obtener modelos más sofisticados.

Si se da un conjunto de datos que representan todos los elementos direccionables de una base de datos, ¿cómo se decide cuáles elementos se deben agrupar juntos en un tipo de registro y qué tipos de registro se deben ligar en conjunto o relaciones para formar una estructura lógica de base de datos? No existe ningún procedimiento específico para el diseño de bases de datos; algunas técnicas son más intuitivas que otras. Por ejemplo, si el contenido de la tabla CUSTOMER-ORDER en la figura 10.1 se debe implantar como base de datos, ¿cuál es la mejor manera de descomponerla para su almacenamiento como base de datos? Evidentemente, toda la tabla, tal y como está, no se debe implantar como una base de datos porque contiene una cantidad excesiva de datos redundantes: la información sobre los clientes se repite cada vez que el cliente hace una orden, y la información del inventario se duplica tantas veces como se vende cada pieza. Para el almacenamiento eficiente y el rápido acceso, la tabla (o relación) se divide en relaciones más pequeñas. Esta separación se puede realizar intuitivamente quitando los datos redundantes del archivo CUSTOMER-ORDER de la siguiente manera:

**PASO 1** En vista que la información de cada cliente se duplica cada vez que el mismo cliente hace algún pedido, estos datos se deben 'factorizar' y sacar fuera para almacenarlos una sola vez en una relación

Archivo CUSTOMER-ORDER

CUST-NO	CUST-NAME	CUST-CITY	UNIT-DELIVERY-FEE	UNIT-PRICE	INV-NO	QTY	DATE (mes/día)
C1	JOHN	OTTAWA	0.75	8.20	I3	1	6/5
C1	JOHN	OTTAWA	0.75	8.20	I3	2	10/12
C2	JANE	TORONTO	1.95	4.00	I2	1	5/15
C2	JANE	TORONTO	1.95	8.20	I3	1	5/15
C2	JANE	TORONTO	1.95	2.00	I1	3	5/15
C3	BOB	OTTAWA	0.75	4.00	I2	1	8/10
C3	BOB	OTTAWA	0.75	2.00	I1	2	10/10
C4	RITA	MONTREAL	1.05	10.50	I4	1	5/05

Agrupación de datos relacionados

Archivo CUST

CUST-NO	CUST-NAME	CUST-CITY
C1	JOHN	OTTAWA
C2	JANE	TORONTO
C3	BOB	OTTAWA
C4	RITA	MONTREAL

Archivo INVENTORY

INV-NO	UNIT-PRICE
I1	2.00
I2	4.00
I3	8.20
I4	10.50

Archivo ORDER

CUST-NO	INV-NO	QTY	DATE
C1	I3	1	6/5
C1	I3	2	10/12
C2	I2	1	5/15
C2	I3	1	5/15
C2	I1	3	5/15
C3	I2	1	8/10
C3	I1	2	10/10
C4	I4	1	5/05

Archivo DELIVERY-FEE

CUST-CITY	UNIT-DELIVERY-FEE
OTTAWA	0.75
TORONTO	1.95
MONTREAL	1.05

FIGURA 10.1 DIVISION DE UN ARCHIVO EN DISTINTOS ARCHIVOS PARA MINIMIZAR LA REDUNDANCIA DE LOS DATOS.

por separado llamada CUST. La información de una pieza del inventario se duplica cada vez que ésta se vende. Por tanto, los datos del inventario se separan en la relación INVENTORY.

**PASO 2** El precio de entrega por unidad (UNIT-DELIVERY-FEE) se repite para cada pedido en la misma ciudad. Por esta razón, los precios UNIT-DELIVERY-FEE se guardan en el archivo DELIVERY-FEE (precio de entrega).

**PASO 3** Las columnas restantes del archivo CUSTOMER-ORDER se guardan en el archivo ORDER.

Como lo muestra la figura, los datos del archivo CUSTOMER-ORDER se agruparon en cuatro registros de tipo conceptual: CUST, ORDER, INVENTORY y DELIVERY-FEE. Los datos redundantes del archivo original CUSTOMER-ORDER se eliminan, excepto para los campos de conexión. Los campos de conexión se usan para ligar CUST con ORDER vía CUST-NO, INVENTORY con ORDER vía INV-NO y DELIVERY-FEE con CUST vía CUST-CITY. Cada uno de los campos de conexión establece una relación entre los datos uno-a-muchos, como lo muestra el modelo correspondiente de la base de datos, figura 10.2. Este modelo está representado por un diagrama de Buchman extendido, aun cuando se usa también otro tipo de diagramas como el de entidad-relación (E-R) o el modelo semántico de datos (Semantic Data Model, SDM). El Modelo consta de cuatro archivos conceptuales, cada uno representa un tipo de entidad. Por ejemplo, la relación CUST contiene varios clientes como entidades, y cada cliente tiene características úni-

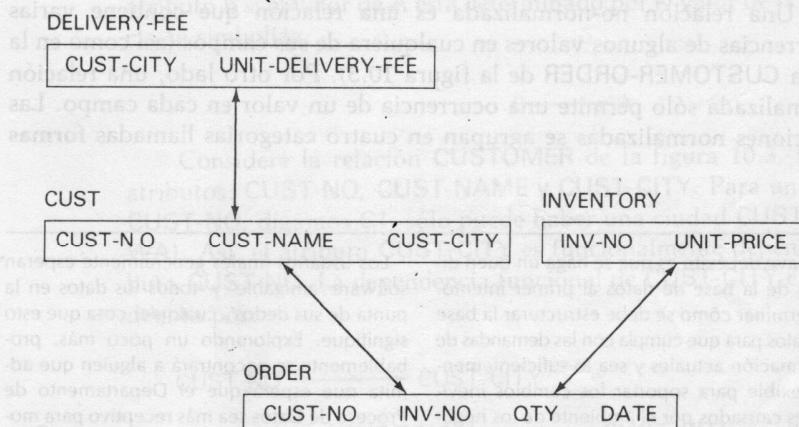


FIGURA 10.2 CONEXIÓN DE DISTINTOS TIPOS DE REGISTROS —OBtenidos de la figura 10.1— EN UN MODELO CONCEPTUAL DE BASE DE DATOS.

cas definidas por un conjunto de atributos: CUST-NO, CUST-NAME y CUST-CITY. Cada entidad se implanta como un registro, y cada atributo corresponde a un campo de un registro.

**NORMALIZACIÓN.** Para el experto diseñador de bases de datos, el derivar entidades o registros de tipo conceptual de un grupo de datos se puede hacer intuitivamente, como lo muestra el ejemplo anterior. Sin embargo, tal intuición no siempre surge espontáneamente en los principiantes, especialmente cuando el diseño es muy complejo. La teoría de la normalización es una ayuda que proporciona un procedimiento riguroso para el diseño de bases de datos. Una base de datos mal diseñada puede funcionar inicialmente, pero puede mostrar anomalías en el almacenamiento debidas al agrupamiento indiscriminado de los campos cuando se efectúen en los archivos las operaciones de inserción, actualización o eliminación. La teoría de la normalización ayuda a reconocer las cualidades no deseadas en un archivo y la forma de corregirlas. (El ejemplo de la figura 10.1 se usará en este capítulo para demostrar que se puede obtener el mismo resultado que el dictado por el sentido común siguiendo una serie de pasos predeterminados en el procedimiento de normalización).

Con el procedimiento de normalización, un archivo conceptual se representa como una tabla de dos dimensiones llamada relación; la forma más simple para representar datos es mediante una tabla. El enfoque relacional generalmente se usa en el proceso de normalización para llegar a modelos de bases de datos; sin embargo, un modelo así derivado se puede modificar para su implantación como DBMS jerárquico, de red o relacional. (Para la terminología básica usada en el enfoque relacional, el lector puede referirse a la sección 5.6.1).

Una relación no-normalizada es una relación que contiene varias ocurrencias de algunos valores en cualquiera de sus campos (así como en la tabla CUSTOMER-ORDER de la figura 10.5). Por otro lado, una relación normalizada sólo permite una ocurrencia de un valor en cada campo. Las relaciones normalizadas se agrupan en cuatro categorías llamadas **formas**

---

La clave del éxito es que se haga un buen diseño de la base de datos al primer intento. Determinar cómo se debe estructurar la base de datos para que cumpla con las demandas de información actuales y sea lo suficientemente flexible para soportar los cambios inevitables causados por el ambiente de los negocios es más un arte que una ciencia.

¿Un ARTE? Sí, poco más o menos. No son sólo 20 reglas fijas a seguir para garantizar el éxito, solamente lineamientos, experiencia y talento.

Los usuarios finales generalmente esperan software 'amigable' y 'todos los datos en la punta de sus dedos', cualquier cosa que esto signifique. Explorando un poco más, probablemente se encontrará a alguien que admite que espera que el Departamento de Proceso de Datos sea más receptivo para modificar las estipulaciones porque 'será muy sencillo hacer los ajustes necesarios'. (¡Haga a un lado a esta persona y asegúrese que le dicen qué están planeando cambiar!)

—Hewlett-Packard (Canada) Ltd.

normales, FN, siendo cada nivel una descomposición más completa de una relación que la del nivel anterior. La meta final del proceso de normalización es la agrupación de todos los atributos (o campos) de una base de datos en relaciones adecuadas para que la base se pueda almacenar con el mínimo de datos redundantes. El propósito de este proceso es quitar las cualidades indeseables de una relación que puedan causar anomalías en el almacenamiento cuando se efectúen operaciones de actualización en la base de datos. El proceso de normalización empieza con la combinación de todos los datos de la base en una relación, la que a su vez se descompone en dos o más relaciones más pequeñas. Se efectúan descomposiciones sucesivas de las relaciones intermedias hasta que todas las relaciones obtenidas pertenecen a la cuarta forma normal (4FN).

Antes de describir completamente el proceso de normalización, se debe descubrir la manera de determinar la forma normal de una relación a partir de su relación de atributos conocida como "dependencia funcional".

## 10.2 DEPENDENCIA FUNCIONAL

El análisis de una relación de dependencia funcional entre los campos de una relación permite clasificar la relación en una de las cuatro formas normales. El concepto de **dependencia funcional** (DF), se tomó de las matemáticas elementales. Se dice que Y es función de X,  $Y = f(X)$ , si el valor de Y está siempre determinado por el valor de X. Si se aplica la misma terminología a una relación, la dependencia funcional entre los atributos A y B en una relación se define como sigue: el atributo A es funcionalmente dependiente del atributo B si el valor de A está determinado por el valor de B. Tal dependencia se simboliza:

$$B \longrightarrow A$$

Considere la relación CUSTOMER de la figura 10.3. Consta de tres atributos: CUST-NO, CUST-NAME y CUST-CITY. Para un valor dado de CUST-NO, digamos C1, sólo puede haber una ciudad CUST-CITY (OTTAWA). Así, el atributo CUST-CITY es funcionalmente dependiente del atributo CUST-NO. La dependencia funcional de CUST-CITY en CUST-NO se denota por:

$$\text{CUST-NO} \longrightarrow \text{CUST-CITY}$$

En otras palabras, CUST-NO determina a CUST-CITY. Se dice que CUST-NO es un **determinante** de CUST-CITY.

Análogamente, el atributo CUST-NAME es funcionalmente dependiente del atributo CUST-NO porque para un valor dado de CUST-NO puede

Relación CUSTOMER

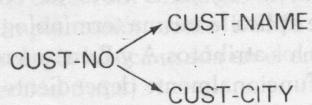
CUST-NO	CUST-NAME	CUST-CITY
C1	JOHN	OTTAWA
C2	JANE	TORONTO
C3	BOB	OTTAWA
C4	RITA	MONTREAL

FIGURA 10.3 DEPENDENCIA FUNCIONAL ENTRE ATRIBUTOS EN UNA RELACION.

Los atributos CUST-NAME y CUST-CITY son funcionalmente dependientes de CUST-NO, porque sólo puede haber un CUST-NAME y un CUST-CITY para un CUST-NO dado.

haber sólo un CUST-NAME. Sin embargo, no se puede decir que CUST-NO es funcionalmente dependiente de CUST-NAME porque más de un CUST-NO se puede asociar con un CUST-NAME si dos o más clientes tienen el mismo nombre.

Los atributos CUST-NAME y CUST-CITY de la relación CUSTOMER son funcionalmente dependientes de su clave principal, CUST-NO. Esta relación de atributos se puede representar por el siguiente diagrama de dependencia funcional:



Se simplificará el diagrama colocando atributos con el mismo determinante en la misma línea:

CUST-NO → CUST-NAME, CUST-CITY

De forma equivalente, el determinante se puede colocar a la derecha de sus atributos dependientes invirtiendo la dirección de la flecha:

CUST-NAME, CUST-CITY ← CUST-NO

Esta representación más simple es útil cuando el diagrama de dependencia funcional contiene una gran cantidad de atributos.

La clave primaria de una relación puede ser una clave compuesta (o concatenada) que consta de más de un campo. Por lo tanto, un atributo puede ser funcionalmente dependiente de un grupo de atributos en vez de sólo un atributo. Un campo en una clave principal compuesta se llama **atributo principal**. Cualquier campo que no forme parte de la clave principal se

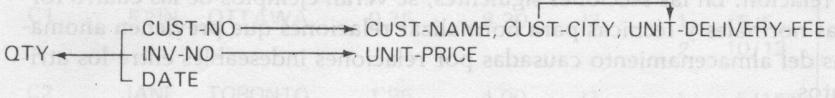
llama **no-clave**. La figura 10.4 ilustra diversos tipos de dependencias funcionales. La relación CUSTOMER-ORDER contiene órdenes de compra del catálogo de clientes de diferentes ciudades. Para este ejemplo se supone:

- A cada artículo ordenado por un cliente se le incluye el precio de manejo y entrega. Esta cantidad está determinada por la ciudad donde el cliente reside.
- Un cliente puede hacer varios pedidos distintos en la misma fecha. Sin embargo, las órdenes de la misma pieza hechas por un solo cliente en la misma fecha, se combinan y las cantidades se suman. Entonces, la clave principal de la relación es combinación de INV-

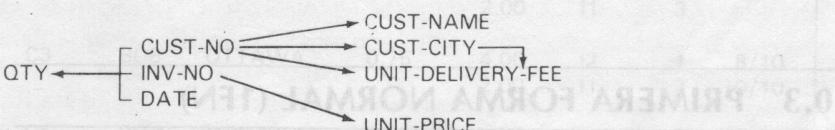
Relación CUSTOMER-ORDER

CUST-NO	CUST-NAME	CUST-CITY	UNIT-DELIVERY-FEE	UNIT-PRICE	INV-NO	QTY	DATE
C1	JOHN	OTTAWA	0.75	8.20	I3	1	6/5
C1	JOHN	OTTAWA	0.75	8.20	I3	2	10/12
C2	JANE	TORONTO	1.95	4.00	I2	1	5/15
C2	JANE	TORONTO	1.95	8.20	I3	1	5/15
C2	JANE	TORONTO	1.95	2.00	I1	3	5/15
C3	BOB	OTTAWA	0.75	4.00	I2	1	8/10
C3	BOB	OTTAWA	0.75	2.00	I1	2	10/10
C4	RITA	MONTREAL	1.05	10.50	I4	1	5/05

a) Contenido de la relación CUSTOMER-ORDER



OR



b) Diagrama de dependencia funcional en la relación. El atributo QTY es total, funcionalmente dependiente de un conjunto de atributos: CUST-NO, INV-NO y DATE, mientras que el atributo no-clave UNIT-DELIVERY-FEE depende de otro atributo no-clave: CUST-CITY.

FIGURA 10.4 DEPENDENCIA FUNCIONAL ENTRE ATRIBUTOS.

NO, CUST-NO y DATE, y su contenido representa de manera única a cada registro de la tabla.

En la práctica, se genera un número único de orden para cada pedido para que las distintas órdenes de un cliente por el mismo artículo y la misma fecha puedan entrar con distintos números de orden. Sin embargo, por la simplicidad del ejemplo, se supondrá que no se generan números de orden y que la combinación de los atributos principales CUST-NO, INV-NO y DATE identifican de manera única cada registro de la relación CUSTOMER-ORDER de la figura 10.4 a).

La figura 10.4 b) muestra dos relaciones equivalentes del diagrama DF para la relación CUSTOMER-ORDER en a). El atributo QTY es funcionalmente dependiente de la clave principal compuesta, CUST-NO, INV-NO y DATE.

El término **dependencia funcional total** se refiere al tipo de dependencia donde un atributo es funcionalmente dependiente de *todos* los campos de la clave primaria en vez de sólo algún subconjunto de la clave primaria. El atributo QTY es totalmente dependiente de la clave primaria porque al valor QTY está determinado por la combinación de los tres atributos principales de esta clave. Sin embargo, la dependencia funcional de UNIT-PRICE en la clave primaria no es total; depende sólo de un subconjunto de esta clave INV-NO. Análogamente, CUST-NAME, CUST-CITY y UNIT-DELIVERY-FEE no son totalmente dependientes de la clave principal porque dependen solamente de CUST-NO.

La figura 10.4 b) muestra además que un atributo no-clave puede ser dependiente de otro atributo no-clave. Por ejemplo, UNIT-DELIVERY-FEE es funcionalmente dependiente de CUST-CITY, que es a su vez no-clave, como lo indica la flecha entre los dos no-clave.

El propósito de establecer un diagrama de dependencia funcional para una relación es usar un método gráfico para determinar la forma normal de la relación. En las secciones siguientes, se verán ejemplos de las cuatro formas normales y técnicas para normalizar relaciones que previenen anomalías del almacenamiento causadas por relaciones indeseables entre los atributos.

### 10.3 PRIMERA FORMA NORMAL (1FN)

Una relación está en la primera forma normal (o 1FN) si todos los campos en cada registro contienen un solo valor tomado de sus dominios respectivos. El dominio de un campo es el rango de valores continuos o discretos permitidos para el campo. Por ejemplo, si los valores del campo QTY son enteros positivos entre 1 y 99, el dominio QTY es el conjunto de enteros 1, 2,...99.

Para ilustrar: la relación de la figura 10.5 no es 1FN porque hay más de un valor contenido en los campos QTY y DATE. Por ejemplo, el campo DATE para la ocurrencia C1 contiene dos valores, 6/5 y 10/12, y su campo QTY también contiene varios valores, 1 y 2. Por lo tanto, la relación no está normalizada y no es un **archivo plano**. (Una relación donde sólo un valor está contenido en cada campo se llama archivo plano).

### 10.3.1 Derivación y anomalías en relaciones 1FN

Una relación sin normalizar se puede normalizar con la creación de un registro nuevo para cada uno de los distintos valores en un campo. Por ejemplo, la figura 10.5 muestra una manera típica con sentido común de guardar la información de los clientes en forma tabular y sin datos duplicados. Pero para que una base de datos funcione uniformemente, se deben crear nuevos registros para que reemplacen los espacios que no contienen algún valor en los campos CUST-NO, CUST-NAME, CUST-CITY, UNIT-DELIVERY-FEE, UNIT-PRICE y INV-NO, con los valores apropiados, como muestra la figura 10.4 a). Esta última relación está en primera forma normal porque cada campo en un registro contiene sólo un valor. En los párrafos siguientes se verá cómo las operaciones de inserción, eliminación o actualización pueden

Relación CUSTOMER-ORDER

CUST-NO	CUST-NAME	CUST-CITY	UNIT-DELIVERY-FEE	UNIT-PRICE	INV-NO	QTY	DATE
C1	JOHN	OTTAWA	0.75	8.20	I3	1	6/5
						2	10/12
C2	JANE	TORONTO	1.95	4.00	I2	1	5/15
				8.20	I3	1	
				2.00	I1	3	
C3	BOB	OTTAWA	0.75	4.00	I2	1	8/10
				2.00	I1	2	10/10
C4	RITA	MONTREAL	1.05	10.50	I4	1	5/05

FIGURA 10.5 EJEMPLO DE UNA RELACION SIN NORMALIZAR.

La relación CUSTOMER-ORDER es un archivo no plano, por lo tanto, es una relación sin normalizar.

causar efectos adversos (o **anomalías de almacenamiento**) en la relación 1FN, y por qué una relación 1FN debe ser objeto de normalizaciones posteriores.

**ANOMALIAS DE INSERCIÓN EN LA RELACION 1FN.** No se puede introducir un nuevo artículo del inventario en la relación CUSTOMER-ORDER en la figura 10.4 a) a menos que el artículo se haya vendido al menos a un cliente. De otra manera los atributos principales CUST-NO y DATE de la clave primaria no estarían definidos; sin embargo, el campo clave-primaria no puede contener un valor nulo o sin definir. Análogamente, no se puede introducir información sobre un cliente nuevo hasta que el cliente haya comprado un artículo.

Las anomalías de inserción en este ejemplo se deben a la dependencia funcional de algunos campos no-clave en un subconjunto de la clave principal en vez de en toda la clave. Como se explicó anteriormente, los atributos CUST-NAME y CUST-CITY son funcionalmente dependientes sólo en el atributo principal CUST-NO. Entonces, si se debe insertar en la relación el registro de un cliente nuevo, INV-NO y DATE quedarán sin definir, situación que no se debe permitir. Para evitar anomalías en la inserción, la información del cliente y del inventario se debe guardar en relaciones separadas.

**ANOMALIAS EN LA ELIMINACION EN LA RELACION 1FN.** Considera el mismo ejemplo de la figura 10.4. Si el artículo 14 del inventario se hace obsoleto y se decide no venderlo, se deben eliminar los registros que contengan el artículo 14. Pero el borrado del registro no sólo quitaría la información del inventario acerca del artículo 14 sino también la información del cliente (Rita) que alguna vez compró este artículo. Del mismo modo, el eliminar la información de un cliente también anularía alguna información valiosa del inventario si es que el artículo fue comprado únicamente por ese cliente. Como en el caso de las anomalías en las inserciones, se pueden evitar las anomalías en la eliminación almacenando la información del cliente y del inventario en relaciones por separado.

**ANOMALIAS DE ACTUALIZACION DE UNA RELACION 1FN.** Cuando una relación sin normalizar se convierte en una relación 1FN, algunos datos se duplican. Tal duplicidad de datos almacenados causará problemas en las operaciones de actualización. Por ejemplo, si la cliente Jane en la figura 10.4 a) se muda de Toronto a Ottawa, se tendrá que modificar cada registro que contenga a Jane; de otra manera se registrarán datos inconsistentes. Es evidente que se necesita un método para eliminar las anomalías de almacenamiento separando la relación 1FN en relaciones más pequeñas.

### 10.3.2 Normalización y verificación de la relación 1FN

Como se describió anteriormente, las anomalías de almacenamiento se pueden atribuir a la presencia de uno o más campos no-clave que no son total y funcionalmente dependientes de la clave principal. Afortunadamente, las anomalías de almacenamiento de la relación IFN se pueden eliminar con el siguiente procedimiento:

- i) quitar de la relación 1FN todos los campos no-clave que no sean totalmente dependientes de la clave primaria.
- ii) guardar los campos no-clave que fueron quitados en relaciones nuevas y adecuadas.

La figura 10.6 muestra la forma en que se identifican y segregan de la relación 1FN CUSTOMER-ORDER, los campos no-clave que causaban problemas, figura 10.4. Primero, hubo que construir un diagrama de dependencia funcional para la relación CUSTOMER-ORDER y así determinar cuáles campos no-clave no son funcionalmente dependientes de la clave principal. Después se separaron estos campos no-clave en nuevas relaciones apropiadas. El procedimiento de la división de una relación en dos o más relaciones más pequeñas en base a las relaciones de atributos se llama **proceso de normalización**. Los detalles son:

**PASO 1** Escoger una clave primaria que pueda representar de manera única cada registro en la relación:

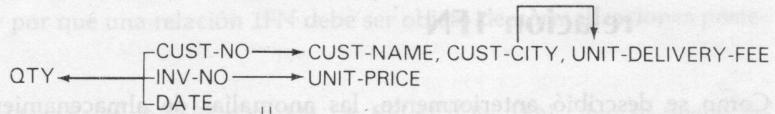
CUSTOMER-ORDER (CUST-NO, INV-NO, DATE, CUST-NAME,  
CUST-CITY, UNIT-DELIVERY-FEE, UNIT-PRICE, QTY)

La clave primaria aparece subrayada. Consta de CUST-NO, INV-NO y DATE.

**PASO 2** Construir un diagrama de dependencia funcional describiendo las relaciones entre los atributos: el diagrama para la relación CUSTOMER-ORDER que se muestra en la figura 10.6 a) indica que sólo QTY es funcionalmente dependiente de una combinación de CUST-NO, INV-NO y DATE. Los atributos CUST-NAME, CUST-CITY y UNIT-DELIVERY-FEE son funcionalmente dependientes de CUST-NO, mientras que UNIT-PRICE es total y funcionalmente dependiente de INV-NO, en vez de serlo de toda la clave principal.

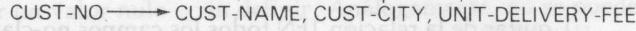
**PASO 3** Dividir la relación 1FN de tal manera que todos los campos no-clave en cada relación dividida sean total y funcionalmente depen-

## Relación CUSTOMER-ORDER



Separación de los atributos no-clave que no son total y funcionalmente dependientes de la clave principal.

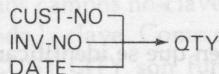
## Relación CUSTOMER



## Relación INVENTORY



## Relación ORDER



- a) La relación 1FN CUSTOMER-ORDER se normaliza separando los atributos no-clave que no son total y funcionalmente dependientes de la clave principal en las relaciones nuevas.

## Relación CUSTOMER

CUST-NO	CUST-NAME	CUST-CITY	UNIT-DELIVERY-FEE
C1	JOHN	OTTAWA	0.75
C2	JANE	TORONTO	1.95
C3	BOB	OTTAWA	0.75
C4	RITA	MONTREAL	1.50

## Relación INVENTORY

INV-NO	UNIT-PRICE
I1	2.00
I2	4.00
I3	8.20
I4	10.50

## Relación ORDER

CUST-NO	INV-NO	DATE	QTY
C1	I3	6/5	1
C1	I3	10/12	2
C2	I2	5/15	1
C2	I3	5/15	1
C2	I1	5/15	3
C3	I2	8/10	1
C3	I1	10/10	2
C4	I4	5/05	1

- b) Contenido de las tres relaciones derivadas de la relación 1FN en a).

FIGURA 10.6 NORMALIZACIÓN DE UNA RELACION 1FN.

dientes de la clave primaria. La relación CUSTOMER-ORDER se descompone en las tres relaciones siguientes:

CUSTOMER (CUST-NO, CUST-NAME, CUST-CITY,  
UNIT-DELIVERY-FEE)  
INVENTORY (INV-NO, UNIT-PRICE)  
ORDER (CUST-NO, INV-NO, DATE, QTY)

El contenido de las relaciones anteriores aparece en la figura 10.6 b).

PRUEBAS PARA LA ELIMINACION DE ANOMALIAS EN 1FN. Las anomalías de inserción, eliminación y actualización asociadas con la relación 1FN se eliminan después de normalizar la relación 1FN. Como muestra la figura 10.6 b), la información acerca de una nueva pieza en el inventario ahora se puede insertar en la relación INVENTORY. También se puede eliminar información sobre los clientes en la relación CUSTOMER sin necesidad de borrar algún dato del inventario. Por último, sólo se debe actualizar un registro, en vez de toda la relación si es que un cliente se cambia de una ciudad a otra.

Una vez que la relación CUSTOMER-ORDER se ha normalizado, el proceso de normalización debe continuar; examinando cada una de las relaciones derivadas (figura 10.6 a)) para buscar otras anomalías (por describir en la siguiente sección). Nuevamente, se construye un diagrama de dependencia funcional para cada relación para determinar a qué forma normal pertenece la relación y decidir si es necesaria una mayor descomposición de las relaciones derivadas.

## 10.4 SEGUNDA FORMA NORMAL (2FN)

Una relación es o pertenece a la segunda forma normal si es 1FN y cada atributo no-clave de la relación es total y funcionalmente dependiente de su clave principal.

Las relaciones derivadas, figura 10.6, son de la segunda forma normal porque todos sus atributos no-clave ya son total y funcionalmente dependientes de sus claves primarias. De cualquier manera, aun cuando las anomalías de almacenamiento 1FN quedan eliminadas cuando se alcanza la 2FN, pueden surgir otro tipo de anomalías de almacenamiento.

### 10.4.1 Anomalías de relaciones 2FN

Una relación 2FN puede presentar anomalías de almacenamiento si cualquiera de sus no-claves **depende transitivamente** de la clave primaria. Se di-

ce que una no-clave depende transitivamente de la clave primaria si es funcionalmente dependiente de otra no-clave, en otras palabras, depende indirectamente de la clave principal. En la figura 10.6 a), el campo **UNIT-DELIVERY-FEE** de la relación **CUSTOMER** es funcionalmente dependiente de **CUST-CITY** en la misma relación. Sin embargo, **CUST-CITY** a su vez es dependiente de **CUST-NO**. Los ejemplos siguientes muestran como esta situación conduce a anomalías de almacenamiento.

**ANOMALIAS DE INSERCIÓN.** El precio de entrega **UNIT-DELIVERY-FEE** para una ciudad en particular no se puede registrar en la relación **CUSTOMER** hasta que haya al menos un cliente que resida en esa ciudad. De otra manera, la clave principal **CUST-NO** quedará sin definir. Esta anomalía se debe a la dependencia funcional del atributo no-clave **UNIT-DELIVERY-FEE** de otro atributo no-clave **CUST-CITY** en la misma relación. De cualquier manera, se pueden evitar tales anomalías de inserción si el atributo **UNIT-DELIVERY-FEE** se registra en otra relación.

**ANOMALIAS DE ELIMINACIÓN.** Suponga que la cliente Rita cierra su cuenta y se elimina el registro correspondiente de la relación **CUSTOMER** (figura 10.6). Como Rita es la única cliente de Montreal, la eliminación del registro también causará que se quite la información **UNIT-DELIVERY-FEE** de \$1.50 para la ciudad de Montreal. Esta anomalía en el borrado se puede atribuir nuevamente a la dependencia funcional del atributo no clave **UNIT-DELIVERY-FEE** con respecto a **CUST-CITY**.

**ANOMALIAS DE ACTUALIZACION.** En la misma relación 2FN **CUSTOMER**, figura 10.6 b), John y Bob son de Ottawa. Si se cambia el precio de entrega **UNIT-DELIVERY-FEE** para Ottawa de \$0.75 a \$1.00, entonces el precio **UNIT-DELIVERY-FEE** se debe actualizar dos veces en vez de sólo una.

#### 10.4.2 Normalización y verificación de una relación 2FN

Las anomalías de almacenamiento en una relación 2FN son causadas por la dependencia transitiva de no-claves en su clave primaria. Por tanto, una de las soluciones al problema es eliminar la dependencia transitiva de la relación 2FN almacenando las no-claves que son transitivamente dependientes de la clave primaria en una relación nueva y adecuada. La relación 2FN **CUSTOMER**, figura 10.7, se puede normalizar como sigue:

**PASO 1** Examinar cada atributo no-clave de la relación **CUSTOMER** para determinar si es funcionalmente dependiente de otra no-clave. El

Diseño de bases de datos: formas normales de relaciones

Relación CUSTOMER  
 CUST-NO → CUST-NAME, CUST-CITY, UNIT-DELIVERY-FEE  
 Eliminación de la dependencia transitiva  
 CUST-NO → CUST-NAME, CUST-CITY

Relación DELIVERY-FEE  
 CUST-CITY → UNIT-DELIVERY-FEE

a) La relación 2FN CUSTOMER se normaliza eliminando el atributo no-clave, transitivamente dependiente UNIT-DELIVERY-FEE, que es funcionalmente dependiente de otra no-clave: CUST-CITY

Relación CUST

CUST-NO	CUST-NAME	CUST-CITY
C1	JOHN	OTTAWA
C2	JANE	TORONTO
C3	BOB	OTTAWA
C4	RITA	MONTREAL

Relación DELIVERY-FEE

CUST-CITY	UNIT-DELIVERY-FEE
OTTAWA	0.75
TORONTO	1.95
MONTREAL	1.50

b) Contenido de las relaciones resultantes de la normalización de la relación 2FN CUSTOMER.

FIGURA 10.7 NORMALIZACION DE UNA RELACION 2FN.

diagrama de dependencia funcional mostrado en la figura 10.7 a) indica que el precio UNIT-DELIVERY-FEE es directamente dependiente de CUST-CITY, pero sólo transitivamente dependiente de CUST-NO.

**PASO 2** Crear una nueva relación para almacenar la no-clave transitivamente dependiente UNIT-DELIVERY-FEE, y su determinante CUST-CITY. La relación 2FN CUSTOMER se divide en dos relaciones, DELIVERY-FEE y CUST. El atributo CUST-CITY debe

aparecer en ambas relaciones para que sirva como campo de conexión para ligar las relaciones **DELIVERY-FEE** y **CUST**. De la figura 10.7 b) se puede determinar que el precio **UNIT-DELIVERY-FEE** para el cliente, digamos **JOHN**, es \$0.75 al asociar el valor **OTTAWA** del campo **CUST-CITY**, que es común a ambas relaciones.

**PRUEBAS PARA LA ELIMINACION DE LAS ANOMALIAS 2FN.** Una vez que desaparece la dependencia transitiva de una relación, se eliminarán las anomalías de la relación 2FN original. Se examinarán ahora las dos relaciones derivadas, en la figura 10.7. Se puede introducir el precio **UNIT-DELIVERY-FEE** para una ciudad aun cuando no haya ningún cliente en la misma. Cuando un cliente desaparece, se puede borrar el registro del cliente sin necesidad de borrar el precio de entrega para la ciudad donde el cliente vive, aún si la persona es el único cliente de esa ciudad. Como el precio **UNIT-DELIVERY-FEE** para cualquier ciudad se almacena sólo una vez, se evita el riesgo de inconsistencia en los datos como resultado de una actualización incompleta de los datos redundantes.

Hasta ahora se ha visto la manera de eliminar anomalías de almacenamiento de relaciones 1FN y 2FN por medio de la normalización. En un proceso de normalización, la relación se divide para remover cierto tipo de dependencias funcionales. Si alguna de las nuevas relaciones derivadas contiene alguna relación indeseable entre sus atributos, continúan las separaciones posteriores hasta que no existan relaciones indeseables en ninguna de las relaciones derivadas.

La figura 10.8 resume los pasos del procedimiento de normalización de la relación **CUSTOMER-ORDER**, figura 10.4. El primer paso involucra una descomposición de la relación 1FN **CUSTOMER-ORDER** para eliminar aquellas no-claves que no son total y funcionalmente dependientes de la clave primaria. Se derivan tres relaciones: **ORDER**, **INVENTORY** y **CUSTOMER**. En el segundo paso, la relación intermedia 2FN **CUSTOMER** se divide aún más para quitar la dependencia transitiva; se obtienen las relaciones **CUST** y **DELIVERY-FEE**. Las relaciones **ORDER** e **INVENTORY** obtenidas del paso uno no contienen dependencias transitivas, y por eso, no se les hace ningún cambio. Como resultado se derivan cuatro relaciones: **ORDER**, **INVENTORY**, **CUST** y **DELIVERY-FEE**. Así, la normalización de la relación **CUSTOMER-ORDER** como se resume en la figura 10.8 logra el mismo resultado que el ejemplo dado en la figura 10.1 donde la relación 1FN se dividió usando el sentido común intuitivo.

## 10.5 TERCERA FORMA NORMAL (3FN)

Una relación es 3FN (tercera forma normal) si es 2FN y ningún atributo no-clave en la relación es funcionalmente dependiente de algún otro atributo no-clave.

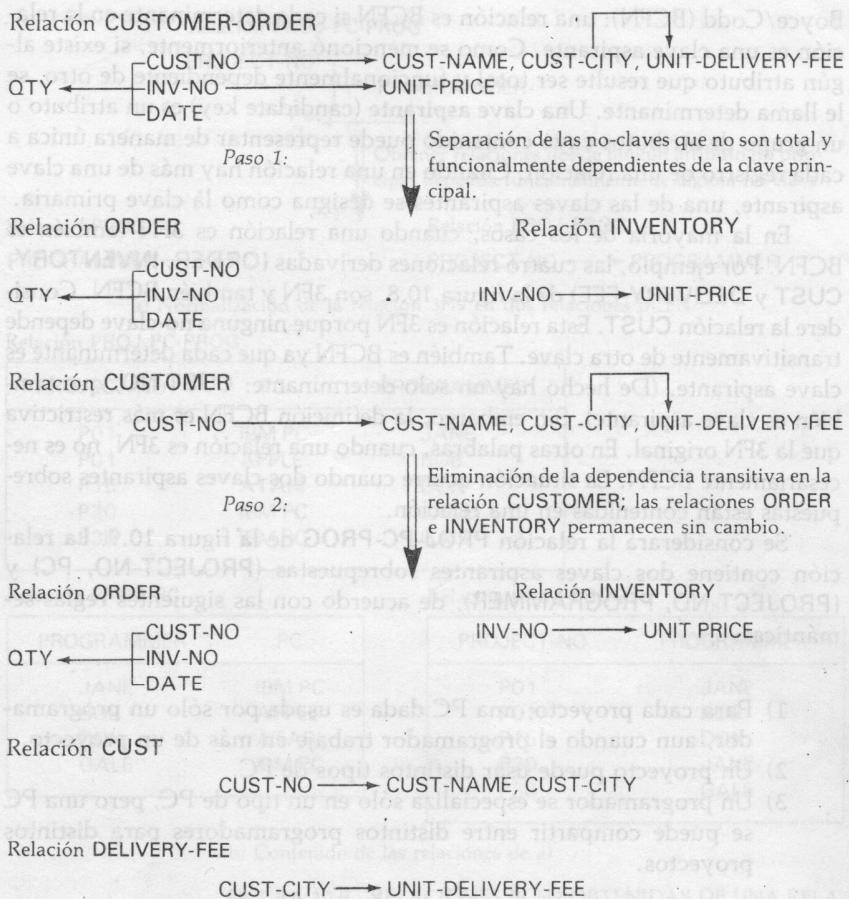


FIGURA 10.8 PASOS DEL PROCEDIMIENTO DE NORMALIZACION

La relación 1FN CUSTOMER-ORDER, mediante el proceso de normalización, se convierte en cuatro relaciones 3FN: ORDER, INVENTORY, CUST y DELIVERY-FEE.

La relación CUSTOMER de la figura 10.6 a) no es 3FN porque UNIT-DELIVERY-FEE es funcionalmente dependiente de un atributo no-clave CUST-CITY. Las cuatro relaciones derivadas (figura 10.8: ORDER, INVENTORY, CUST, DELIVERY-FEE) son 3FN. En la mayoría de los casos el proceso de normalización queda completo cuando todas las relaciones derivadas son 3FN.

Codd dio la definición original de 3FN en 1972. Se le corrigió posteriormente y la definición revisada 3NF se conoce como la forma normal

Boyce/Codd (BCFN): una relación es BCFN si cada determinante en la relación es una clave aspirante. Como se mencionó anteriormente, si existe algún atributo que resulte ser total y funcionalmente dependiente de otro, se le llama determinante. Una clave aspirante (candidate key) es un atributo o un grupo de atributos cuyo contenido puede representar de manera única a cada registro de una relación. Cuando en una relación hay más de una clave aspirante, una de las claves aspirantes se designa como la clave primaria.

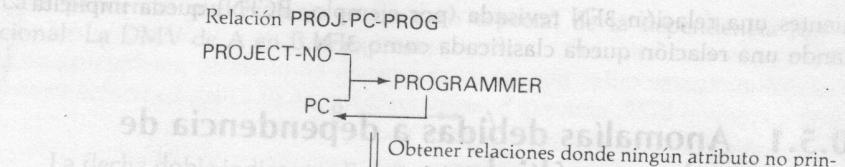
En la mayoría de los casos, cuando una relación es 3FN también es BCFN. Por ejemplo, las cuatro relaciones derivadas (ORDER, INVENTORY, CUST y DELIVERY-FEE) de la figura 10.8, son 3FN y también BCFN. Considere la relación CUST. Esta relación es 3FN porque ninguna no-clave depende transitivamente de otra clave. También es BCFN ya que cada determinante es clave aspirante. (De hecho hay un solo determinante: CUST-NO, que también es clave aspirante). Sin embargo, la definición BCFN es más restrictiva que la 3FN original. En otras palabras, cuando una relación es 3FN, no es necesariamente BCFN. La situación ocurre cuando dos claves aspirantes sobrepuertas están contenidas en una relación.

Se considerará la relación PROJ-PC-PROG de la figura 10.9. La relación contiene dos claves aspirantes sobrepuertas (PROJECT-NO, PC) y (PROJECT-NO, PROGRAMMER), de acuerdo con las siguientes reglas semánticas:

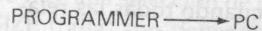
- 1) Para cada proyecto, una PC dada es usada por sólo un programador, aun cuando el programador trabaje en más de un proyecto.
- 2) Un proyecto puede usar distintos tipos de PC.
- 3) Un programador se especializa sólo en un tipo de PC, pero una PC se puede compartir entre distintos programadores para distintos proyectos.

Si se designa una de las claves aspirantes, digamos PROJECT-NO, PC), como la clave principal, el diagrama de dependencia funcional de la figura 10.9 a) resume las reglas semánticas anteriores. El diagrama muestra un tipo especial de dependencia cuando el atributo principal PC es dependiente de la no-clave PROGRAMMER.

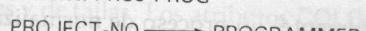
La relación PROJ-PC-PROG es 3FN porque ninguna no-clave es transitivamente dependiente de otras no-claves y la no-clave es totalmente dependiente de la clave primaria. Sin embargo, la relación no es BCFN ya que PROGRAMMER, que no es clave aspirante, es un determinante de PC. Como resultado, la relación sufre de ciertas anomalías de almacenamiento. Por ejemplo, si se contrata a un nuevo programador para trabajar con la APPLE, no es posible insertar el registro en la relación PROJ-PC-PROG, figura 10.9 b), hasta que se dé un proyecto al programador. La dificultad aquí es que el atributo principal PROJECT-NO no puede quedar sin definir. Otro ejemplo de anomalía de almacenamiento sucede cuando se desea eliminar la información sobre el proyecto P15. Tal operación también quitaría la



Relación PROG-PC



Relación PROJ-PROG



a) Normalización de la relación 3FN en dos relaciones BCFN

Relación PROJ-PC-PROG

PROJECT-NO	PC	PROGRAMMER
P01	IBM PC	JANE
P01	APPLE	BOB
P15	ATARI	DON
P20	IBM PC	JANE
P30	IBM PC	GALE

Relación PROG-PC

PROGRAMMER	PC
JANE	IBM PC
BOB	APPLE
DON	ATARI
GALE	IBM PC

Relación PROJ-PROG

PROJECT-NO	PROGRAMMER
P01	JANE
P01	BOB
P15	DON
P20	JANE
P30	GALE

b) Contenido de las relaciones de a)

FIGURA 10.9 RELACIONES BCFN OBTENIDAS DE UNA RELACIÓN 3FN.

información que Don es especialista en ATARI porque P15 es el único proyecto en que trabaja Don.

Las anomalías 3FN de la relación PROJ-PC-PROG suceden cuando hay dos claves sobreuestas y cuando uno de los atributos principales depende funcionalmente de una no-clave. El problema puede superarse proyectando la relación en dos relaciones BCFN, como se muestra en la figura 10.9 a). La parte b) muestra el contenido de las relaciones en a), donde puede comprobarse si las relaciones derivadas BCFN (PROG-PC y PROJ-PROG) eliminan las anomalías de almacenamiento de la relación 3FN, PROJ-PC-PROG.

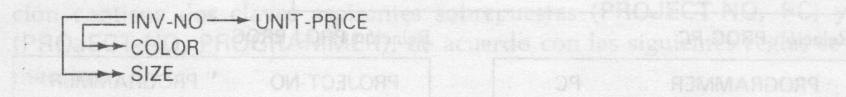
Del ejemplo anterior se puede ver que la definición de 3FN no es una guía suficiente para la normalización cuando una relación contiene un atributo principal dependiente de una no-clave. Sin embargo, la definición original de 3FN es adecuada en la mayoría de los casos. En los ejemplos si-

guientes una relación 3FN revisada (por ejemplo, BCFN) queda implícita cuando una relación queda clasificada como 3FN.

### 10.5.1 Anomalías debidas a dependencia de valores múltiples

Generalmente, un proceso de normalización termina cuando todas las relaciones derivadas pertenecen a la tercera forma normal. Sin embargo, si una relación contiene dependencias de valores múltiples, es necesaria una normalización posterior. Dada una relación, el atributo A de esta relación se dice ser **dependiente de multivalores** (DMV) del atributo B si un rango específico de valores del atributo A está determinado por un valor particular de B.

Relación INVENTORY



a) Diagrama de dependencia con atributos de dependencia funcional y multivalor

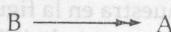
INV-NO	UNIT-PRICE	COLOR	SIZE
I1	2.00	BLUE	1
I1	2.00	BLUE	2
I1	2.00	BLUE	3
I1	2.00	BLUE	4
I1	2.00	RED	1
I1	2.00	RED	2
I1	2.00	RED	3
I1	2.00	RED	4
I2	4.00	PINK	1
I2	4.00	PINK	2
I2	4.00	GREEN	1
I2	4.00	GREEN	2
I2	4.00	RED	1
I2	4.00	RED	2

b) Contenido de la relación INVENTORY

FIGURA 10.10 DEPENDENCIA MULTIVALOR.

Se dispone de cada artículo del inventario en distintos colores y tallas. La clave principal de la relación anterior es INV-NO, COLOR y SIZE.

La dependencia multivalores es un caso especial de la dependencia funcional. La DMV de A en B se expresa:



La flecha doble indica que B define valores múltiples en A. Considérese la relación INVENTORY de la figura 10.10 b). Para cada artículo del inventario hay una selección de distintos colores y de tallas. Los atributos COLOR y SIZE (talla) son dependientes multivalores de INV-NO. En otras palabras, cualquier valor de INV-NO determina un rango de valores en los campos SIZE y COLOR. El precio unitario UNIT-PRICE es total y funcionalmente dependiente de INV-NO: para artículos con el mismo número INV-NO, el precio unitario permanece constante sin importar el color o la talla. Si el precio variara para ciertas tallas, se deberían de asignar distintos números de inventario. Por ejemplo, los zapatos tallas uno a cuatro pueden tener el mismo precio bajo un solo número INV-NO, mientras que los zapatos tallas cinco a ocho pueden tener un precio distinto y estar clasificados bajo otro número INV-NO.

La clave principal para la relación INVENTORY en la figura 10.10 a) consta de INV-NO, COLOR y SIZE. El atributo INV-NO no puede identificar de manera única a cada registro, porque bajo un número dado de INV-NO existen diversos artículos con diferentes colores y tallas. Por lo tanto, los atributos dependientes multivalores COLOR y SIZE debe pertenecer a la clave principal.

El problema evidente con la dependencia multivalores es la redundancia de datos. Como muestra la figura 10.10 b), el artículo del inventario 11 se repite ocho veces ya que viene en cuatro tallas y dos colores. Esta redundancia se agrava cuando la mayoría de las piezas están disponibles en varias tallas y colores.

### 10.5.2 Tratamiento de la dependencia de valores múltiples

La redundancia de los datos causada por la dependencia multivalores se puede eliminar siguiendo uno de los dos siguientes métodos:

- i) crear una nueva relación para cada atributo DMV
- ii) reemplazar un atributo DMV con atributos funcionalmente dependientes (DF)

Cada método se ilustra con un ejemplo.

#### 1) Creación de una relación para cada atributo DMV

Cuando el número de valores repetidos en un DMV es muy grande, se puede crear una nueva relación para el atributo DMV y su clave principal.

Considerando el ejemplo de la figura 10.11, si suponemos que el número de tallas para un artículo del inventario es muy grande o indefinido (esto es, pueden agregarse tallas nuevas en el futuro), entonces se puede crear una nueva relación **SIZE**, como se muestra en la figura 10.11 b). La relación contiene todas las tallas disponibles para cada **INV-NO**.

### 2) Reemplazando un atributo DMV con varios atributos DF

Si la cantidad de valores distintos en un atributo DMV es un número específico y pequeño, entonces cada uno de los valores del atributo DMV se puede representar por un atributo dentro del mismo registro. Por ejemplo, si la cantidad de colores para cualquier **INV-NO** dado no excede de tres, el atributo **COLOR** se puede representar por **COLOR1**, **COLOR2**, **COLOR3** como se muestra en la figura 10.11 b). Cuando un artículo del inventario está disponible sólo en dos colores, se le puede asignar a **COLOR3** un valor nulo o indefinido.

La diferencia básica entre los dos tratamientos está en la manera en que se almacenan los distintos valores para un atributo DMV. Con el primer método se crea una nueva relación para un atributo DMV, y los distintos valores que ocurren en un campo DMV se agregan verticalmente —como registros— a la nueva relación. En el segundo método, los distintos valores occurrentes se agregan horizontalmente en un número fijo de campos, por ejemplo, el atributo multivalor **COLOR** se representa por medio de tres campos. La limitante de este método es que no se reserva espacio para un número indefinido de colores. Por otro lado, el primer método permite la futura expansión de nuevos valores insertándolos como nuevos registros. El precio de conseguir flexibilidad para acomodar nuevos valores del atributo se pagó con la creación de una nueva relación y el almacenamiento redundante del campo de conexión; por ejemplo, **INV-NO** en la relación **SIZE**.

### 10.5.3 Cuarta forma normal (4FN)

Una relación es 4FN (cuarta forma normal) si es BCFN y no contiene dependencias multivalores.

Una manera de entender una DMV no-trivial consiste en explicar primero qué es una DMV trivial. Sea **R** una relación con campos **A**, **B**, **C**. La relación se llama trivial si una combinación de valores **A** y **B** determina los valores de **C**. La DMV se expresa:

$$(A, B) \longrightarrow C$$

En otras palabras, una combinación de **A** y **B** multidetermina a **C**.

Usando la misma relación **R** (**A,B,C**), ocurre una dependencia multivalor **no trivial** cuando **B** es dependiente multivalor de **A**, y **C** también es dependiente multivalor de **A**:

## Relación INVENTORY



Se crea la nueva relación SIZE para el atributo DMV SIZE.

## Relación INV-COLOR



El atributo multivalor COLOR en la relación INV-COLOR se reemplaza por 3 atributos dependientes: COLOR1, COLOR2 y COLOR3. SIZE permanece sin cambios.

## Relación INV

$INV\text{-}NO \rightarrow UNIT\text{-}PRICE, COLOR1, COLOR2, COLOR3$

## Relación SIZE

$INV\text{-}NO \rightarrow SIZE$

- Se crea una nueva relación SIZE para el atributo DMV SIZE y su determinante INV-NO, mientras que los distintos colores de una pieza del inventario pueden representarse dentro de cada tupla por tres atributos: COLOR1, COLOR2 y COLOR3.

## Relación INV

INV-NO	UNIT-PRICE	COLOR1	COLOR2	COLOR3
I1	2.00	BLUE	RED	-
I2	4.00	PINK	GREEN	RED

## Relación SIZE

INV-NO	SIZE
I1	1
I1	2
I1	3
I1	4
I2	1
I2	2

- contenido de las relaciones resultantes SIZE e INV.

FIGURA 10.11 TRATAMIENTO DE LOS ATRIBUTOS DEPENDIENTES MULTIVALORES.



Así, una relación debe contener una clave compuesta que consiste al menos en tres campos subclave antes que pueda ocurrir una DMV no trivial. Por tanto, una relación 3FN con clave primaria con menos de tres atributos principales es automáticamente 4FN. (Una dependencia multivalor no trivial también es llamada dependencia unión-binaria no trivial).

Considere nuevamente la relación INVENTORY, figura 10.10. Esa relación no es 4FN, porque tiene una dependencia multivalor no trivial (o DMV unión-binaria no trivial):



Como se muestra ahí, el problema evidente con DMV no triviales es la excesiva duplicidad de datos. La figura 10.12 a) muestra esquemáticamente la forma de eliminar los DMV no triviales: la relación R se descompone en R1 (A, B) y R2 (A, C). La figura 10.12 b) regresa al ejemplo concreto de eliminar una relación MVD no trivial de la relación INVENTORY. Cada uno de los dos atributos MVD se separa para crear las nuevas relaciones COLOR y SIZE. El resultado del proceso de normalización obtenido de este ejemplo difiere del de la figura 10.11 porque se supuso que el atributo DMV COLOR en este ejemplo también contiene una cantidad ilimitada de valores anidados que se repiten. Por lo tanto, en este ejemplo se creó una nueva relación COLOR en vez de usar COLOR1, COLOR2 y COLOR3 para reemplazar el atributo DMV COLOR. Se puede ver del ejemplo en la figura 10.12 que el tratamiento de una DMV no trivial es equivalente al tratamiento de dos DMV individuales.

En circunstancias normales no se requiere más normalización después de 3FN, porque ya se han tomado en cuenta los atributos DMV durante la etapa inicial de la normalización para derivar un archivo plano 1FN. La figura 10.13 resume las etapas del proceso de normalización.

## 10.6 EL PROCESO DE NORMALIZACION EN EL DISEÑO DE BASES DE DATOS

La primera fase para el diseño de una base de datos —llegar el modelo conceptual de la base de datos— involucra el **proceso iterativo** de agrupar los datos relacionados dentro de relaciones. Mediante distintas interpretaciones de las reglas semánticas, el procedimiento de normalización se repite y se re-

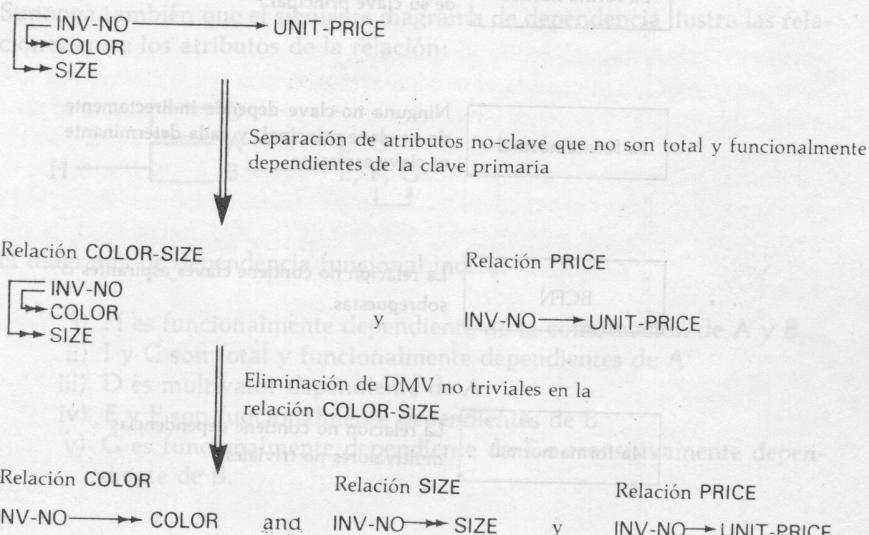
Eliminación de dependencia multivalor no trivial

R a cabo el proceso de normalización para eliminar no triviales (vease figura)

$\boxed{A \rightarrow B}$   $\rightarrow A \rightarrow B$  y  $A \rightarrow C$

- a) La relación R con dependencias multivalores se descompone en dos relaciones 4FN: R1 y R2.

#### Normalización de la relación INVENTORY



- b) La relación INVENTORY se normaliza para obtener tres relaciones 4FN: COLOR, SIZE y PRICE.

FIGURA 10.12 DERIVACION DE RELACIONES 4FN ELIMINANDO DEPENDENCIAS MULTIVALORES NO TRIVIALES.

quiere de diversas pruebas con distintas agrupaciones antes de llegar al diseño satisfactorio. El proceso de normalización utiliza una técnica formal y rigurosa para agrupar finalmente los datos en relaciones 4FN. Aunque otros autores han usado distintos **métodos sintéticos de diseño**, la técnica de normalización descrita en este capítulo ofrece los lineamientos más sencillos a seguir por diseñadores principiantes y gerentes. Sin embargo, inicialmente también se pueden usar otros enfoques para llegar al modelo propuesto, que luego se puede verificar por medio del método **analítico** de normalización.

## Sin normalizar

Datos almacenados redundantemente en un archivo no plano.

Así, una relación se considera sin normalizar si se almacena en tres campos su clave ante-

## la forma normal

La relación tiene registros por separado pa-

ra cada valor en cada campo del registro, o

cada campo de un registro contiene un solo

valor.

Considérese nuevamente la relación *INVENTORY*, figura 10.10. Esta rela-

## 2a forma normal

Cada clave depende total y funcionalmente

de su clave principal.

## 3a forma normal

Ninguna no-clave depende indirectamente

de su clave principal, y cada determinante

es clave aspirante.

## BCFN

La relación no contiene claves aspirantes

sobrepuertas.

## 4a forma normal

La relación no contiene dependencias

multivalores no triviales.

FIGURA 10.13 NIVELES DE NORMALIZACION.

El proceso de normalización descompone una relación inicial universal conteniendo todos los datos del diseño de la base en varias relaciones más pequeñas. Cada una de las relaciones resultantes se examina para ver si es necesaria una normalización posterior. De cualquier modo, de las relaciones derivadas finales, se puede reconstruir cualquier información de las relaciones original o intermedias. La descomposición de una relación en varias relaciones normalizadas se dice que es **sin pérdida**, porque cualquier relación intermedia se puede reconstruir de las relaciones 4FN finales mediante operaciones **JOIN** y **PROJECT** (descritas en el capítulo 7).

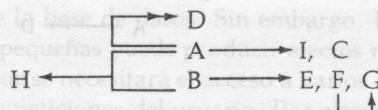
El proceso de normalización es un sistema seguro para el diseño de bases de datos. Sin embargo, el diseñador debe entender el significado de todos los datos de la base y de las dependencias interrelacionadas antes de que el proceso de normalización pueda empezar. El determinar las relaciones

entre los atributos depende del sentido común del diseñador y del entendimiento a fondo del medio ambiente de la base de datos. Los pasos generales para llevar a cabo el proceso de normalización son los siguientes (véase figura 10.14):

**PASO 1** Determinar todas las relaciones de dependencia funcional y multivalores entre los atributos. Suponga para esta explicación, que los atributos globales de la base por diseñar están incluidos en la siguiente relación universal:

Relación R1 (A, B, C, D, E, F, G, H, I)

Suponga también que el siguiente diagrama de dependencia ilustra las relaciones entre los atributos de la relación:



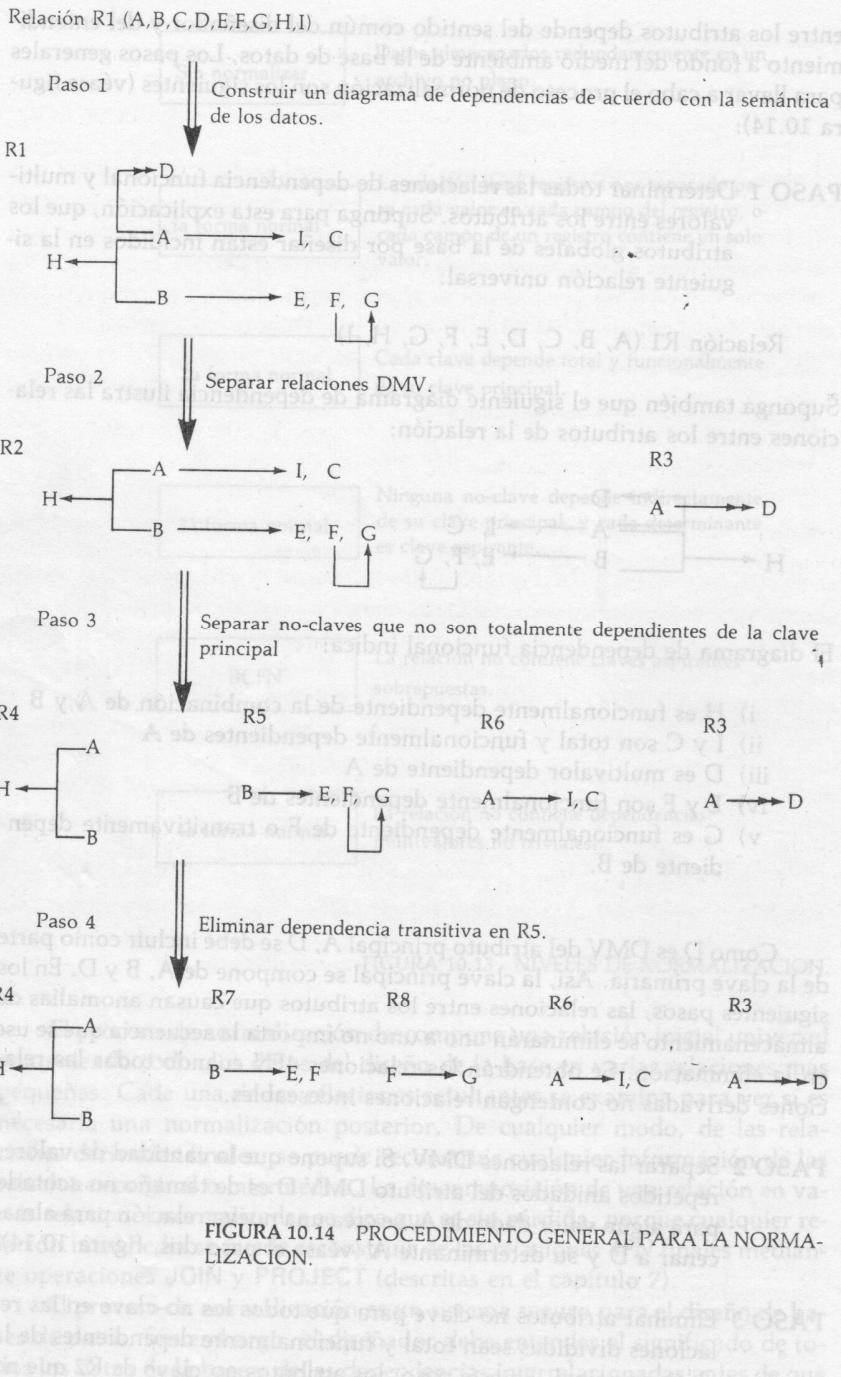
El diagrama de dependencia funcional indica:

- H es funcionalmente dependiente de la combinación de A y B
- I y C son total y funcionalmente dependientes de A
- D es multivalor dependiente de A
- E y F son funcionalmente dependientes de B
- G es funcionalmente dependiente de F o transitivamente dependiente de B.

Como D es DMV del atributo principal A, D se debe incluir como parte de la clave primaria. Así, la clave principal se compone de A, B y D. En los siguientes pasos, las relaciones entre los atributos que causan anomalías de almacenamiento se eliminarán uno a uno no importa la secuencia que se use en la eliminación. Se obtendrán las relaciones 4FN cuando todas las relaciones derivadas no contengan relaciones indeseables.

**PASO 2** Separar las relaciones DMV. Si supone que la cantidad de valores repetidos anidados del atributo DMV D es de tamaño no acotado para algún valor dado de A, se crea una nueva relación para almacenar a D y su determinante A (véase el paso dos, figura 10.14).

**PASO 3** Eliminar atributos no-clave para que todos los no-clave en las relaciones divididas sean total y funcionalmente dependientes de la clave principal. En este paso, los atributos no-clave de R2 que no son totalmente dependientes de la clave principal se dividen en



nuevas relaciones apropiadas, y se derivan tres nuevas relaciones más chicas R4, R5, R6. La relación R3 permanece sin cambios.

**PASO 4** Separar las no-claves transitivamente dependientes. El atributo no-clave G en R5 depende de otro atributo no-clave, F. Entonces, R5 sufre una descomposición 'sin pérdida' en dos relaciones, R7 y R8. Las relaciones R4, R6 y R3 permanecen sin cambio. El resultado de normalizar R1 son las cinco relaciones mostradas en el último renglón de la figura 10.14. Todas son 4FN porque cada determinante es clave aspirante y no existe dependencia multivaluada no trivial.

A pesar que algunos campos se puedan duplicar en las relaciones normalizadas (aquejlos usados como campos de conexión para enlazar relaciones), la normalización generalmente implica la reducción del almacenamiento de la base de datos. Sin embargo, la división de una relación en varias más pequeñas puede producir efectos negativos en el desempeño de la base porque se necesitará el acceso a varios archivos antes que puedan contestarse las peticiones del usuario. Por ejemplo, compare la rapidez de recuperación en la relación 1FN, CUSTOMER-ORDER y en sus cuatro relaciones 4FN derivadas, figura 10.8, con la siguiente consulta:

*Listar los nombres de los clientes que compraron al menos un artículo del inventario con precio UNIT-PRICE de \$8.20*

Con la relación 1FN sólo es necesario el acceso a un archivo para responder la consulta. En el caso de las relaciones 4FN, hay que que accesar tres tipos de registro distintos (INVENTORY, CUST y ORDER) para satisfacer la misma consulta. En contraste, una operación de actualización que cambie el precio unitario de un artículo del inventario será mucho más rápida con las relaciones 4FN que con la relación IFN porque los datos redundantes en la relación 1FN ya han sido factorizados en la relación separada INVENTORY.

El objetivo del proceso de normalización es minimizar las anomalías encontradas en las operaciones de inserción, eliminado y actualización. Sin embargo, para minimizar las anomalías de almacenamiento hay que pagar a cambio con el desempeño del sistema. A veces es deseable recombinar ciertas relaciones 4FN para mejorar el desempeño del sistema de base de datos.

## 10.7 RESUMEN

- El enfoque analítico usado en el proceso de normalización empieza con una relación universal inicial, que incluye todos los atributos de la base de datos por diseñar. La relación universal y cualquier relación derivada

de ésta estará sujeta a divisiones sucesivas cuando se encuentren una o más de las condiciones siguientes en la relación:

- i) la relación contiene uno más atributos DMV
- ii) uno o más atributos no-clave no son funcionalmente dependientes de la clave principal
- iii) uno o más atributos no-clave de la relación dependen funcionalmente de otra no-clave, o no todos los determinantes de la relación son claves aspirantes.

El proceso de normalización implica una descomposición sin pérdida de una relación en varias relaciones más pequeñas. Esto es, la información contenida en la relación universal o en cualquiera de sus relaciones intermedias se puede reconstruir a partir de las relaciones finales.

- Una relación se clasifica como no normalizada, 1FN, 2FN, 3FN, o 4FN de la siguiente manera:
  - i) Un archivo no plano es una relación sin normalizar donde uno o más atributos contienen valores anidados repetidos.
  - ii) Una relación es 1FN si cada uno de sus atributos contiene un solo valor.
  - iii) Una relación es 2FN si es 1FN y todos sus atributos no-clave son total y funcionalmente dependientes de la clave principal.
  - iv) Una relación es 3FN si es 2FN y ningún atributo no-clave es transitivamente dependiente de la clave principal.
  - v) La definición revisada de 3FN se llama BCFN. Es más restrictiva que la definición original de 3FN en que todos los determinantes en una relación BCFN deben ser claves aspirantes.
  - vi) Una relación es 4FN cuando es 3FN y no contiene ninguna dependencia multivaluada no trivial.

La normalización después de la tercera forma normal implica que se trata de una dependencia multivalor no trivial, o MVD unión-binaria. En la realidad es rara vez necesaria la normalización más allá de la tercera forma normal, porque la dependencia multivalor en general se trata en la etapa inicial del proceso de normalización.

- El tratamiento de un atributo DMV necesita generalmente de la creación de una nueva relación para que contenga el atributo DMV y su determinante. Sin embargo, si el número máximo de valores repetidos de un atributo DMV es un número pequeño, el atributo DMV se puede reemplazar por un número fijo de atributos funcionalmente dependientes para que todas las ocurrencias de sus valores se puedan representar en el mismo registro.
- El separar o dividir una relación para eliminar anomalías de almacenamiento puede tener el impacto negativo en el desempeño de la base de

datos porque se pueden requerir operaciones extra de Entrada/Salida para recuperar datos de distintos archivos. Así, para decidir el grado de normalización deseado, el diseñador debe considerar cuidadosamente los tipos de operaciones que se efectuarán con más frecuencia.

## REFERENCIAS

- Date, C.J., *An Introduction to Database Systems*, Vol. 1. Addison-Wesley, 1981.  
Hawryszkiewycz, I.T., *Database Analysis and Design*, SRA Inc., 1984.  
Kroenke, D.M., *Database Processing: Fundamentals, Design, Implementation*, SRA Inc., 1983.

## Preguntas de repaso y ejercicios

1. Defina cada uno de los siguientes términos mediante un ejemplo:
  - a) dependencia funcional, dependencia funcional total, dependencia transitiva.
  - b) relación sin normalizar, 1FN, 2FN, 3FN, BCNF, 4FN.
  - c) dependencia multivalor, DMV no trivial.
2. Describa dos métodos distintos para eliminar datos redundantes causados por un atributo DMV en una relación.
3. Comente los efectos de la normalización en el almacenamiento y el desempeño de la base de datos.
4. (Este problema se puede adaptar para su uso en un taller de trabajo). Use el procedimiento de normalización para diseñar un modelo conceptual de base de datos para el sistema de inventarios de una compañía que tiene tiendas en distintos lugares del país. Los atributos representados en la base de datos son:

TIENDA-NUMERO  
TIENDA-NOMBRE  
TIENDA-CIUDAD  
TIENDA-ESTADO  
INV-NUMERO  
INV-NOMBRE  
INV-CANTIDAD  
TAMAÑO  
COLOR  
PRECIO-UNITARIO  
IMPUESTO-VENTA

## Ejemplo Resuelto 4FN explicado detalladamente

### Formalización CERO

usuarios				
nombre	empresa	direccion_empresa	url1	url2
Joe	ABC	1 Work Lane	abc.com	xyz.com
Jill	XYZ	1 Job Street	abc.com	xyz.com

Diríamos que la anterior tabla esta en nivel de Formalizacion Cero porque ninguna de nuestras reglas de normalización ha sido aplicada. Observa los campos url1 y url2 -- ¿Qué haremos cuando en nuestra aplicación necesitemos una tercera url? ¿Quieres tener que añadir otro campo/columna a tu tabla y tener que reprogramar toda la entrada de datos de tu código PHP? Obviamente no, tu quieres crear un sistema funcional que pueda crecer y adaptarse fácilmente a los nuevos requisitos. Hechemos un vistazo a las reglas del Primer Nivel de Formalización-Normalización, y las aplicaremos a nuestra tabla.

Primer nivel de Formalización/Normalización. (F/N):

1. Eliminar los grupos repetitivos de las tablas individuales.
2. Crear una tabla separada por cada grupo de datos relacionados.
3. Identificar cada grupo de datos relacionados con una clave primaria.

¿Ves que estamos rompiendo la primera regla cuando repetimos los campos url1 y url2? ¿Y que pasa con la tercera regla, la clave primaria?

La regla tres básicamente significa que tenemos que poner un campo tipo contador auto-incrementable para cada registro. De otra forma, ¿Qué pasaría si tuviéramos dos usuarios llamados Joe y queremos diferenciarlos? Una vez que aplicáramos el primer nivel de F/N nos encontrariamos con la siguiente tabla:

usuarios				
userId	nombre	empresa	direccion_empresa	url
1	Joe	ABC	1 Work Lane	abc.com
1	Joe	ABC	1 Work Lane	xyz.com
2	Jill	XYZ	1 Job Street	abc.com
2	Jill	XYZ	1 Job Street	xyz.com

Ahora diremos que nuestra tabla está en el primer nivel de F/N. Hemos solucionado el problema de la limitación del campo url. Pero sin embargo vemos otros problemas. Cada vez que introducimos un nuevo registro en la **tabla usuarios**, tenemos que duplicar el nombre de la empresa y del usuario. No sólo nuestra BD crecerá muchísimo, sino que será muy facil que la BD se corrompa si escribimos mal alguno de los datos redundantes. Aplicaremos pues el segundo nivel de F/N:

## Segundo nivel de F/N

1. Crear tablas separadas para aquellos grupos de datos que se aplican a varios registros.
2. Relacionar estas tablas mediante una clave externa.

Hemos separado el campo url en otra tabla, de forma que podemos añadir más en el futuro si tener que duplicar los demás datos. También vamos a usar nuestra clave primaria para relacionar estos campos:

usuarios			
userId	nombre	empresa	direccion_empresa
1	Joe	ABC	1 Work Lane
2	Jill	XYZ	1 Job Street
urls			
urlId	relUserId	url	

1	1	abc.com
2	1	xyz.com
3	2	abc.com
4	2	xyz.com

Vale, hemos creado tablas separadas y la clave primaria en la **tabla usuarios**, userId, está relacionada ahora con la clave externa en la **tabla urls**, relUserId. Esto está mejor. ¿Pero qué ocurre cuando queremos añadir otro empleado a la empresa ABC? ¿o 200 empleados? Ahora tenemos el nombre de la empresa y su dirección duplicándose, otra situación que puede inducirnos a introducir errores en nuestros datos. Así que tendrémos que aplicar el tercer nivel de F/N.

## Tercer nivel de F/N.

1. Eliminar aquellos campos que no dependan de la clave.
- Nuestro nombre de empresa y su dirección no tienen nada que ver con el campo **userId**, así que tienen que tener su propio empresaId:

usuarios		
userId	nombre	relEmpresaId
1	Joe	1
2	Jill	2
empresas		
emprId	empresa	direccion_empresa
1	ABC	1 Work Lane
2	XYZ	1 Job Street
urls		
urlId	RelUserId	url
1	1	abc.com
2	1	xyz.com
3	2	abc.com
4	2	xyz.com

Ahora tenemos la clave primaria emprId en la **tabla empresas** relacionada con la clave externa recEmpresaId en la **tabla usuarios**, y podemos añadir 200 usuarios mientras que sólo tenemos que insertar el nombre 'ABC' una vez.

Nuestras tablas de usuarios y urls pueden crecer todo lo que quieran sin duplicación ni corrupción de datos.

La mayoría de los desarrolladores dicen que el tercer nivel de F/N es suficiente, que nuestro esquema de datos puede manejar fácilmente los datos obtenidos de una cualquier empresa en su totalidad, y en la mayoría de los casos esto será cierto.

Pero hechemos un vistazo a nuestro campo urls - ¿Ves duplicación de datos? Esto es perfectamente aceptable si la entrada de datos de este campo es solicitada al usuario en nuestra aplicación para que teclee libremente su url, y por lo tanto es sólo una coincidencia que Joe y Jill teclearon la misma url. ¿Pero qué pasa si en lugar de entrada libre de texto usáramos un menú desplegable con 20 o incluso más urls predefinidas? Entonces tendríamos que llevar nuestro diseño de BD al siguiente nivel de F/N, el cuarto, muchos desarrolladores lo pasan por alto porque depende mucho de un tipo muy específico de relación, la relación 'varios-con-varios', la cual aún no hemos encontrado en nuestra aplicación.

### Relaciones entre los Datos

Antes de definir el cuarto nivel de F/N, veremos tres tipos de relaciones entre los datos: uno-a-uno, uno-con-varios y varios-con-varios. Mira la **tabla usuarios** en el Primer Nivel de F/N del ejemplo de arriba. Por un momento imaginemos que ponemos el campo url en una tabla separada, y cada vez que introducimos un registro en la **tabla usuarios** también introducimos una sola fila en la **tabla urls**. Entonces tendríamos una relación uno-a-uno: cada fila en la tabla usuarios tendría exactamente una fila correspondiente en la **tabla urls**.

Para los propósitos de nuestra aplicación no sería útil la normalización. Ahora mira las tablas en el ejemplo del Segundo Nivel de F/N. Nuestras tablas permiten a un sólo usuario tener asociadas varias urls. Esta es una relación uno con- varios, el tipo de relación más común, y hasta que se nos presentó el dilema del Tercer Nivel de F/N. la única clase de relación que necesitamos.

La relación varios-con-varios, sin embargo, es ligeramente más compleja. Observa en nuestro ejemplo del Tercer Nivel de F/N que tenemos a un usuario relacionado con varias urls. Como dijimos, vamos a cambiar la estructura para permitir que varios usuarios estén relacionados con varias urls y así tendremos una relación varios-con-varios. Veamos como quedarían nuestras tablas antes de seguir con este planteamiento:

usuarios		
userId	nombre	relEmpresaId
1	Joe	1
2	Jill	2
empresas		
empriId	empresa	direccion_empresa
1	ABC	1 Work Lane

2	XYZ	1 Job Street
urls		
urlId	url	
1	abc.com	
2	xyz.com	
url_relations		
relationId	relatedUrlId	relatedUserId
1	1	1
2	1	2
3	2	1
4	2	2

Para disminuir la duplicación de los datos (este proceso nos llevará al Cuarto Nivel de F/N), hemos creado una tabla que sólo tiene claves externas y primarias **url\_relations**. Hemos sido capaces de remover las entradas duplicadas en la tabla urls creando la tabla **url\_relations**. Ahora podemos expresar fielmente la relación que ambos Joe and Jill tienen entre cada uno de ellos, y entre ambos, las urls.

Así que veamos exáctamente que es lo que el Cuarto Nivel de F/N. supone:

#### Cuarto Nivel de F/N.

1. En las relaciones varios-con-varios, entidades independientes no pueden ser almacenadas en la misma tabla.

Ya que sólo se aplica a las relaciones varios-con-varios, la mayoría de los desarrolladores pueden ignorar esta regla de forma correcta. Pero es muy útil en ciertas situaciones, tal

como está. Hemos optimizado nuestra **tabla urls** eliminando duplicados y hemos puesto las relaciones en su propia tabla.

1º Se presenta la siguiente tabla de restaurantes de pizza, el tipo y el lugar de envío:

N- Restaurante	N-Tipo de pizza	N-Lugar de envío
Pizza Hut	Americana	Surco
Pizza Hut	Hawaiana	Miraflores
Papa Johns	Americana	San Borja
Papa Johns	Hawaiana	San Luis

Se puede observar que hay dependencia multivalor entre el nombre del restaurante y el número de pizza, por ello se aplica la cuarta forma normal que daría paso a dos tablas:

N- Restaurante	N-Tipo de pizza 1	N-Tipo de pizza 2	N- Restaurante	N-Lugar de envío
Pizza Hut	Americana	Hawaiana	Pizza Hut	Surco
Papa Johns	Americana	Hawaiana	Papa Johns	Miraflores
Papa Johns	Americana	Hawaiana	Papa Johns	San Borja
Papa Johns	Americana	Hawaiana	Papa Johns	San Luis

## Ejercicios resueltos

### Primer ejemplo.

### Aplicar las reglas de normalización los siguientes ejercicios.

1. Un dato sin normalizar no cumple con ninguna regla de normalización. Para explicar con un ejemplo en qué consiste cada una de las reglas, vamos a considerar los datos de la siguiente tabla.

**ordenes** (id\_orden, fecha, id\_cliente, nom\_cliente, estado, num\_art, nom\_art, cant, precio)

### Ordenes

Id_orden	Fecha	Id_cliente	Nom_cliente	Estado	Num_art	nom_art	cant	Precio
2301	23/02/11	101	Martin	Caracas	3786	Red	3	35,00
2301	23/02/11	101	Martin	Caracas	4011	Raqueta	6	65,00
2301	23/02/11	101	Martin	Caracas	9132	Paq-3	8	4,75
2302	25/02/11	107	Herman	Coro	5794	Paq-6	4	5,00

2303	27/02/11	110	Pedro	Maracay	4011	Raqueta	2	65,00
2303	27/02/11	110	Pedro	Maracay	3141	Funda	2	10,00

### Segundo ejemplo:

Tenemos una empresa pública donde los puestos de trabajo están regulados por el Estado, de modo que las condiciones salariales están determinadas por el puesto. Se ha creado el siguiente esquema relacional

EMPLEADOS(nss, nombre, puesto, salario, emails) con nss como clave primaria.

nss	nombre	puesto	salario	emails
111	Juan Pérez	Jefe de Área	3000	juanp@ecn.es; jefe2@ecn.es
222	José Sánchez	Administrativo	1500	jsanchez@ecn.es
333	Ana Diaz	Administrativo	1500	adiaz@ecn.es; ana32@gmail.com
...	...	...	...	...

### Tercer ejemplo.

Tenemos un control de certificados en cual queremos ordenar de manera mas clara y precisa realiza la normalización de la tabla dada aplicando el criterio correspondiente.

A	B	C	D	E	F	G	H	I	J	
1 CONTROL DE CERTIFICADOS										
2										
3										
4	NO.EMP	NOMBRE_E	APPELLIDO_E	NO.CERTIFICA	CVE_TIPO	TIPO_CERTIFICA	F_EXPEDICION	MAT_ESTU	NOM_ESTU	CARRERA
5	1567 NELLY	ANGEL		513	1 PARCIAL	12-jul-12	901144 JUAN	ICO		
6	1567 NELLY	ANGEL		514	2 COMPLETO	13-jul-12	901144 JUAN	ICO		
7	1567 NELLY	ANGEL		515	1 PARCIAL	15-jul-13	902245 LUIS	ISC		
8	1567 NELLY	ANGEL		516	2 COMPLETO	18-sep-13	902245 LUIS	ISC		
9	1567 NELLY	ANGEL		517	3 PROMEDIO/SEM	18-sep-13	691124 MARITZA	ISC		
10	1567 NELLY	ANGEL		518	3 PROMEDIO/SEM	18-sep-13	789066 GABRIELA	ICO		
11	1789 DIANA	LOPEZ		519	3 PROMEDIO/SEM	18-sep-13	103066 YARITZA	ICO		
12	1789 DIANA	LOPEZ		520	3 PROMEDIO/SEM	20-sep-13	112345 JUAN	ICO		
13	1789 DIANA	LOPEZ		521	3 PROMEDIO/SEM	20-sep-13	114523 PEDRO	ISC		
14	1789 DIANA	LOPEZ		522	3 PROMEDIO/SEM	20-sep-13	662345 YAIR	ICO		
15	1789 DIANA	LOPEZ		523	3 PROMEDIO/SEM	20-sep-13	565656 JOSE	ICO		
16	1890 ERIKA	CONTRERAS		567	2 COMPLETO	20-sep-13	556677 JOSE	ISC		
17	1890 ERIKA	CONTRERAS		568	2 COMPLETO	20-sep-13	334455 ADRIANA	ICO		
18	1550 GABRIELA	OROZCO		569	2 COMPLETO	20-sep-13	223344 RUBI	ISC		

## Cuarto ejercicio de normalización

En el siguiente problema tenemos el control de internos de un cerezo donde nos proporcionan información tanto de interno como de policías que en el radican en este caso aplicaremos la normalización para llevar un mejor control del sistema .

## Quinto ejercicio de normalización

En este caso se lleva el control de una escuela el cual realiza certificaciones para los alumnos ellos nos están dando los siguientes datos que ellos manejan debamos ordenar la información y tratar de aplicar la normalización correspondiente para tener la información más ordenada y detallada.

## Sexto Ejercicio de Normalización

En las siguientes tablas el usuario nos proporciona la siguiente información sobre el control de autobuses en el cual se realizan préstamos al cliente y también llevan el control de a quienes se les ofrece el servicio lo que se requiere es aplicar la normalización para que llevar un mejor control de información.

M13										
	A	B	C	D	E	F	G	H	I	J
1										
2	<b>AUTOBUSES</b>									
3	<b>NO_AUTOBUS</b>	<b>MARCA</b>	<b>TIPO</b>	<b>PLAZAS</b>						
4	1	MERCEDES	PASAJEROS	41						
5	2	TOYOTA	PASAJEROS	15						
6	3	MERCEDES	PASAJEROS	35						
7										
8										
9	<b>PRESTAMO</b>									
10	<b>NO_AUTOBUS</b>	<b>F_PRESTAMO</b>	<b>F_INICIO</b>	<b>HORA_INICIO</b>	<b>F_FIN</b>	<b>HORA_FIN</b>	<b>ORIGEN</b>	<b>DESTINO</b>	<b>CVE_PERSONA</b>	
11	1	10-dic-12	15-dic-12	06:00	10-ene-13	16:00	CARMEN	CANCUN	22	
12	2	10-may-12	01-jun-12	05:00	05-jun-12	12:00	CARMEN	MERIDA	23	
13	3	15-abr-12	10-may-12	05:00	10-may-12	19:00	CARMEN	SABANCUY	28	
14										
15										
16	<b>RESPONSABLE</b>				<b>FACULTAD</b>					
17	<b>CVE_PERSONA</b>	<b>NOM_PERSONA</b>	<b>APE_PERSONA</b>	<b>FACULTAD</b>	<b>ID_FACULTAD</b>	<b>NOM_FACULTAD</b>				
18	22	JULIAN	MONTERO	1	1	NATURALES				
19	23	PEDRO	PICENT	2	2	MATEMATICAS				
20	28	ANDRÉS	SALAZAR	3	3	INFORMATICA				
21										

## Ejercicios a resolver

### Ejercicio 2.1 Relación Inventario Cables

RELACIÓN INVENTARIO_CABLES									
CODIGO	NOMBRE	CALIBRE	COLOR	P.U.	STOCK	F_ALTA	No_NOMINA	NOM_NOMIN	APE_NOMINA
1122	WIRE	8 AWG	BLACK	\$ 15.00	500	01/11/2014	513	JULIAN	CABRERA
1122	WIRE	8 AWG	WHITE	\$ 15.00	500	01/11/2014	513	JULIAN	CABRERA
1122	WIRE	8 AWG	RED	\$ 15.00	100	31/10/2014	628	ANA	SAENZ
1122	WIRE	8 AWG	GREEN	\$ 15.00	100	31/10/2014	628	ANA	SAENZ
1123	WIRE	10 AWG	BLACK	\$ 12.00	200	01/11/2014	513	JULIAN	CABRERA
1123	WIRE	10 AWG	WHITE	\$ 12.00	200	01/11/2014	513	JULIAN	CABRERA
1123	WIRE	10 AWG	RED	\$ 12.00	200	31/10/2014	628	ANA	SAENZ
1123	WIRE	10 AWG	GREEN	\$ 12.00	200	31/10/2014	628	ANA	SAENZ
1124	WIRE	12 AWG	BLACK	\$ 10.00	200	01/11/2014	513	JULIAN	CABRERA
1124	WIRE	12 AWG	WHITE	\$ 10.00	200	01/11/2014	513	JULIAN	CABRERA
1124	WIRE	12 AWG	RED	\$ 10.00	100	31/10/2014	628	ANA	SAENZ
1124	WIRE	12 AWG	GREEN	\$ 10.00	100	31/10/2014	628	ANA	SAENZ
1125	WIRE	14 AWG	BLACK	\$8.00	100	01/11/2014	513	JULIAN	CABRERA
1125	WIRE	14 AWG	WHITE	\$8.00	100	01/11/2014	513	JULIAN	CABRERA
1125	WIRE	14 AWG	RED	\$8.00	100	31/10/2014	628	ANA	SAENZ
1125	WIRE	14 AWG	GREEN	\$8.00	150	31/10/2014	628	ANA	SAENZ

## Ejercicio 2.2 Tienda Virtual Tennis

TIENDA_VIRTUAL_TENNIS														
NUMCL	NOMBRECL	APELLIDOCL	DIRECCIONCL	NUMPEDIDO	PARTIDA	CANTIDAD	MODELO	TALLA	COLOR	UNITARIO	FECHAPEDIDO	FECHAENTRE		
25	ALBERTO	AGUILERA	MANIGUA	100	1	1	RONALDO	28	NEGRO	500	29/04/2014	08/05/2014		
25	ALBERTO	AGUILERA	MANIGUA	100	2	1	NIKE	28	BLANCO	600	29/04/2014	08/05/2014		
25	ALBERTO	AGUILERA	MANIGUA	100	3	1	REEBOOK	28	BLANCO	700	30/04/2014	09/05/2014		
25	ALBERTO	AGUILERA	MANIGUA	100	4	1	MESSI	28	BARCELONA	600	30/04/2014	09/05/2014		
26	OSCAR	GARRIDO	FRANCISCO I	101	1	2	NIKE	29	BLANCO	600	29/04/2014	08/05/2014		
26	OSCAR	GARRIDO	FRANCISCO I	101	2	2	NIKE	29	NEGRO	500	29/04/2014	08/05/2014		
26	OSCAR	GARRIDO	FRANCISCO I	101	3	2	NIKE	29	AZUL	400	30/04/2014	09/05/2014		
26	OSCAR	GARRIDO	FRANCISCO I	101	4	2	NIKE	29	ROJO	400	30/04/2014	09/05/2014		
26	OSCAR	GARRIDO	FRANCISCO I	101	5	2	NIKE	29	VERDE	500	30/04/2014	09/05/2014		

### Ejercicio 2.3 Relación Ventas Zapaterías

Relación Vtas_Zapaterías									
CVE_VTA	PARTIDA	CVE_PROD	MODELO	TALLA	CANTIDAD	P.U.	PRODUCTO	ALMACEN	
111	1	12	1122	27	1	450	CABALLERO	5	
111	2	13	3344	22	1	400	TEEN	8	
111	3	13	3344	24	1	400	TEEN	10	
111	4	15	2211	25	1	450	DAMA	15	
112	1	12	1123	26	2	400	CABALLERO	10	
113	1	15	2212	23	2	400	DAMA	10	
NULL	NULL	12	1122	26	NULL	NULL	CABALLERO	6	
NULL	NULL	12	1122	25	NULL	NULL	CABALLERO	8	
NULL	NULL	12	1122	24	NULL	NULL	CABALLERO	8	
NULL	NULL	15	1123	23	NULL	NULL	DAMA	6	
NULL	NULL	15	1123	24	NULL	NULL	DAMA	6	
NULL	NULL	15	1123	25	NULL	NULL	DAMA	6	
NULL	NULL	13	3344	23	NULL	NULL	TEEN	7	
NULL	NULL	13	3344	21	NULL	NULL	TEEN	8	
NULL	NULL	13	3355	21	NULL	NULL	TEEN	8	
NULL	NULL	13	3366	22	NULL	NULL	TEEN	8	
NULL	NULL	13	3366	23	NULL	NULL	TEEN	8	

## Ejercicio 2.4 Relación Ventas Material Eléctrico

RELACIÓN VENTAS_MATERIAL_ELECTRICO											
CODIGO	NOMBRE	CALIBRE	COLOR	P.U.	F_VENTA	H_VENTA	CANTIDAD	No_FACTURA	NOM_FACTUR	APE_FACTURA	
1122	WIRE	8 AWG	BLACK	\$ 150.00	01/11/2014	10:00	5	9945	ROSARIO	BAÑOS	
1122	WIRE	8 AWG	WHITE	\$ 150.00	01/11/2014	10:00	5	9945	ROSARIO	BAÑOS	
1122	WIRE	8 AWG	RED	\$ 150.00	31/10/2014	10:15	1	9980	ULISES	LOPES	
1122	WIRE	8 AWG	GREEN	\$ 150.00	31/10/2014	10:15	1	9980	ULISES	LOPES	
1123	WIRE	10 AWG	BLACK	\$ 120.00	01/11/2014	10:00	3	9945	ROSARIO	BAÑOS	
1123	WIRE	10 AWG	WHITE	\$ 120.00	01/11/2014	10:00	3	9945	ROSARIO	BAÑOS	
1123	WIRE	10 AWG	RED	\$ 120.00	31/10/2014	10:15	1	9980	ULISES	LOPES	
1123	WIRE	10 AWG	GREEN	\$ 120.00	31/10/2014	10:15	1	9980	ULISES	LOPES	
1124	WIRE	12 AWG	BLACK	\$ 100.00	01/11/2014	10:00	2	9945	ROSARIO	BAÑOS	
1124	WIRE	12 AWG	WHITE	\$ 100.00	01/11/2014	10:00	2	9945	ROSARIO	BAÑOS	
1124	WIRE	12 AWG	RED	\$ 100.00	31/10/2014	10:15	1	9980	ULISES	LOPES	
1124	WIRE	12 AWG	GREEN	\$ 100.00	31/10/2014	10:15	1	9980	ULISES	LOPES	
1125	WIRE	14 AWG	BLACK	\$80.00	01/11/2014	10:15	1	9980	ULISES	LOPES	
1125	WIRE	14 AWG	WHITE	\$80.00	01/11/2014	10:15	1	9980	ULISES	LOPES	
1125	WIRE	14 AWG	RED	\$80.00	31/10/2014	10:00	1	9945	ROSARIO	BAÑOS	
1125	WIRE	14 AWG	GREEN	\$80.00	31/10/2014	10:00	1	9945	ROSARIO	BAÑOS	

## Ejercicio 2.5 Relación Kardex

Relación KARDEX								
Matricula	Nom_estudiante	Ape_estudiante	Cve_materia1	Materias	Calificacion	Semestres	ID_PE	NOM_PE
901144	Eduardo	Torres	100-1	Bases de datos I	80	1	1	ISC
901144	Eduardo	Torres	100-2	Bases de datos II	90	2	1	ISC
901144	Eduardo	Torres	100-3	Bases de datos III	80	3	1	ISC
901144	Eduardo	Torres	200-1	BD Distribuidas	80	4	1	ISC
904523	Martín	Montejo	100-1	Bases de datos I	85	1	1	ISC
904523	Martín	Montejo	100-2	Bases de datos II	90	2	1	ISC
904523	Martín	Montejo	100-3	Bases de datos III	90	3	1	ISC
904523	Martín	Montejo	200-1	BD Distribuidas	80	4	1	ISC
904523	Martín	Montejo	200-2	BD Multimedia	90	5	1	ISC
901144	Eduardo	Torres	300-1	Ingeniería de SW	80	1	1	ISC
901144	Eduardo	Torres	400-1	Circuitos lógicos	80	1	1	ISC
901144	Eduardo	Torres	500-1	Circuitos combinacionales	80	1	1	ISC

## Ejercicio 2.6 Relación Profesores Proyectos

Relación PROFESORES_PROYECTOS												
No.Emp	NomEmpleado	ApEmpleado	Cve_proy1	NomProyecto1	MontoProyecto1	Num_Informe	Nom_Informe	Num_informe	Nom_informe	Num_informe	Nom_informe	...
1234	Patricia	Zavaleta	513	ROA	950,000	1	Reporte Parcial	2	Reporte Parcial	3	Reporte Parcial	...
1255	Gustavo	Verduzco	514	Redes	999,000	1	Reporte Parcial	2	Reporte Parcial	3	Reporte Parcial	...
1588	Victor	Hernández	516	Android	590,000	1	Reporte Parcial	2	Reporte Parcial	3	Reporte Parcial	...
1688	Eduardo	Orbe	517	BlackBerry	590,000	1	Reporte Parcial	2	Reporte Parcial	3	Reporte Parcial	...
1455	Eduardo	Torres	518	iOS	590,000	1	Reporte Parcial	2	Reporte Parcial	3	Reporte Parcial	...
Num_informe	Nom_informe	MatEstudiant	NomEstudiant	MatEstudiante2	NomEstudiante2	MatEstudiante	NomEstudiante3					
4	Reporte Final	3312	Juan Antonio	3315	Rubi del Carmen	3355	Luis Miguel					
4	Reporte Final	5512	Wilson	2255	Gabriela	6677	Cruz					
4	Reporte Final	8812	Irvin	1233	Carlos	8811	Esmeralda					
4	Reporte Final	9988	Enri	1243	Zafiro	8814	Diamantina					
4	Reporte Final	6678	Raquel	1278	Rubi	8815	Jade					

## Ejercicio 2.7 Relación Óptica

OPTIMA									
<u>CVE</u>	<u>TIPO</u>	<u>COLOR</u>	<u>PRECIO</u>	<u>STOCK</u>	<u>ID GENERO</u>	<u>GENERO</u>	<u>CVE VENTA</u>		<u>CANTIDAD</u>
100	POLO	WHITE	80	200	1	FEMENINO	1222		5
101	POLO	BLACK	100	250	1	FEMENINO	1222		3
102	POLO	RED	100	200	1	FEMENINO	1322		1
103	POLO	YELLOW	80	100	1	FEMENINO	1344		1
104	POLO	PINK	100	200	1	FEMENINO	1355		1
104	POLO	PINK	100	200	1	FEMENINO	1425		1
104	POLO	PINK	100	200	1	FEMENINO	1416		1
104	POLO	PINK	100	200	1	FEMENINO	1427		1
200	POLO	WHITE	80	200	2	MASCULINO	7866		2
200	POLO	WHITE	80	200	2	MASCULINO	7899		2
201	POLO	BLACK	100	250	2	MASCULINO	66		2
201	POLO	BLACK	100	250	2	MASCULINO	77		2
202	POLO	RED	100	200	2	MASCULINO	88		2
202	POLO	RED	100	200	2	MASCULINO	99		2
203	POLO	YELLOW	80	100	2	MASCULINO	22		2
204	POLO	PINK	100	200	2	MASCULINO	33		2
300	SPORT	BLANCO	30	100	2	MASCULINO	-		-
300	SPORT	GRIS	30	100	2	MASCULINO	-		-
300	SPORT	NEGRO	30	100	2	MASCULINO	-		-

## Ejercicio 2.8 Cliente óptica

OPTICAS															
NUMCL	NOMBRECL	APELLIDOCL	CVE_ARMA	TIPO_ARMAZON	GRADUAIZQ	GRADUADERECHA	COLOR	TIPO	MARCA	PEDIDO	CVE TIENDA	NOM TIENDA	SUC TIENDA	DIRECCION	
1	KASSANDRA	CERINO	100	POLICARBONATO	0.5	0.5	CAFÉ	CASUAL	JEAN	1	1	OPTICAS	MATRIZ	PLAZA PALMIRA	
1	KASSANDRA	CERINO	100	POLICARBONATO	0.5	0.5	NEGRO	SPORT	JEAN	1	2	OPTICAS	SUCURSAL 1	PLAZA CAMARON	
1	KASSANDRA	CERINO	100	POLICARBONATO	0.5	0.5	ROJO	GERENCIAL	JEAN	1	3	OPTICAS	SUCURSAL 2	PLAZA AMERICAS	
2	OSCAR	GARRIDO	101	FIBRACARBONO	0.25	0.25	NEGRO	SPORT	RAY BAN	23	1	OPTICAS	MATRIZ	PLAZA PALMIRA	
2	OSCAR	GARRIDO	101	FIBRACARBONO	0.25	0.25	CAFÉ	CASUAL	RAY BAN	24	2	OPTICAS	SUCURSAL 1	PLAZA CAMARON	
3	YAIR	ARGENTE	200	CAREY	0.5	1	NEGRO	CASUAL	POLICE	30	3	OPTICAS	SUCURSAL 2	PLAZA AMERICAS	
3	YAIR	ARGENTE	200	CAREY	0.5	1	CAFÉ	CASUAL	POLICE	31	1	OPTICAS	MATRIZ	PLAZA PALMIRA	

## **Clave de respuestas**

## **Referencias:**

Libro Sistema de base de Datos: Administración Y Uso  
Autor Alice Y.H.Tsai  
Segunda edición

Libro Base De Datos (Desde Che Hasta Codd Con Oracle)  
Autores: Irene Luque Ruiz  
Miguel Ángel Gómez –Nieto  
Gonzalo Cerruela García  
Dpto. Informática y Análisis Numérico  
Universidad de Córdoba