Egibide

Unidad Didáctica 4:

NORMALIZACIÓN DE LAS RELACIONES

1. Problemas del esquema relacional

Una vez obtenido el esquema relacional resultante del modelo entidad relación que representaba la base de datos, normalmente tendremos una buena base de datos. Pero otras veces, debido a fallos en el diseño o a problemas indetectables en esta fase del diseño, tendremos un esquema que puede producir una base de datos que incorpore estos problemas:

- **Redundancia.** Se llama así a los datos que se repiten continua e innecesariamente por las tablas de las bases de datos.
- **Ambigüedades.** Datos que no clarifican suficientemente el registro al que representan.
- **Pérdida de restricciones de integridad.**
- **Anomalías en operaciones de modificación de datos.** El hecho de que al insertar un solo elemento haya que repetir tuplas en una tabla para variar unos pocos datos. O que eliminar un elemento suponga eliminar varias tuplas.

En la figura 1 se muestra una tabla que almacena datos sobre los profesores que imparten asignaturas. Si observamos con atención esta tabla, vemos que presenta varios de los problemas enumerados anteriormente:

Cod_Prof	Nombre_ Prof	Despacho	Cod_asignatura	Nombre_ asignatura	Nº créditos
9321	J. Sánchez	2B2	ITIG0231	Diseño de BD	7
9321	J. Sánchez	2B2	ITIG0221	Ficheros y BD	7
8142	P. Martín	2B2	ITIG0231	Diseño de BD	7
8142	P. Martín	2B2	ITIG0242	Diseño Avanzado de BD	4,5
8142	P. Martín	2B2	LD0241	SGBD	6
9577	A. García	2C4	II0232	Administración de BD	6
9111	L. López	2D5	II0232	Administración de BD	6
9111	L. López	2D5	ITIG0232	BD Avanzadas	4,5
9111	L. López	2D5	ITIG0221	Ficheros y BD	7

Figura 1. Ejemplo de diseño inadecuado tabla IMPARTE

El principio fundamental reside en que las tablas deben referirse a objetos o situaciones muy concretas. Lo que ocurre es que conceptualmente es difícil obtener ese problema. La solución suele ser dividir la tabla con problemas en otras tablas más adecuadas.

2. Proceso de normalización

El proceso de **normalización** de una base de datos consiste en aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo E-R (entidad-relación) al modelo relacional que elimine las dependencias no deseadas entre los atributos.

Las bases de datos relacionales se normalizan para:

- Evitar la **redundancia** de los datos.
- Evitar **problemas de actualización** (tras insertar, modificar o borrar) de los datos en las tablas.
- Proteger la **integridad** de los datos.

La normalización se lleva a cabo en una serie de pasos, llamados **formas normales**, que van reconstruyendo las tablas haciéndolas más robustas y menos vulnerables a las anomalías que pudiesen surgir tras una actualización. Las formas normales se corresponde a una teoría de normalización iniciada por el propio Codd y continuada por otros autores (entre los que destacan Boyce y Fagin). Codd definió en 1970 la primera forma normal, desde ese momento aparecieron la segunda, tercera, la Boyce-Codd, la cuarta y la quinta forma normal.

- 1ª FN (Codd, 1970)
  Concepto de relación normalizada.
- 2ª, 3ª FN (Codd, 1970), FNBC (Boyce/Codd, 1974)
  Basadas en análisis de dependencias funcionales.
- 4ª FN. Fagin, 1977
  Basada en análisis de dependencias multivaluadas.
- 5ª FN. Fagin, 1979
  Basada en análisis de dependencias de proyección / combinación.

Una tabla puede encontrarse en primera forma normal y no en segunda forma normal, pero no al contrario. Es decir los números altos de formas normales son más restrictivos (la quinta forma normal cumple todas las anteriores).

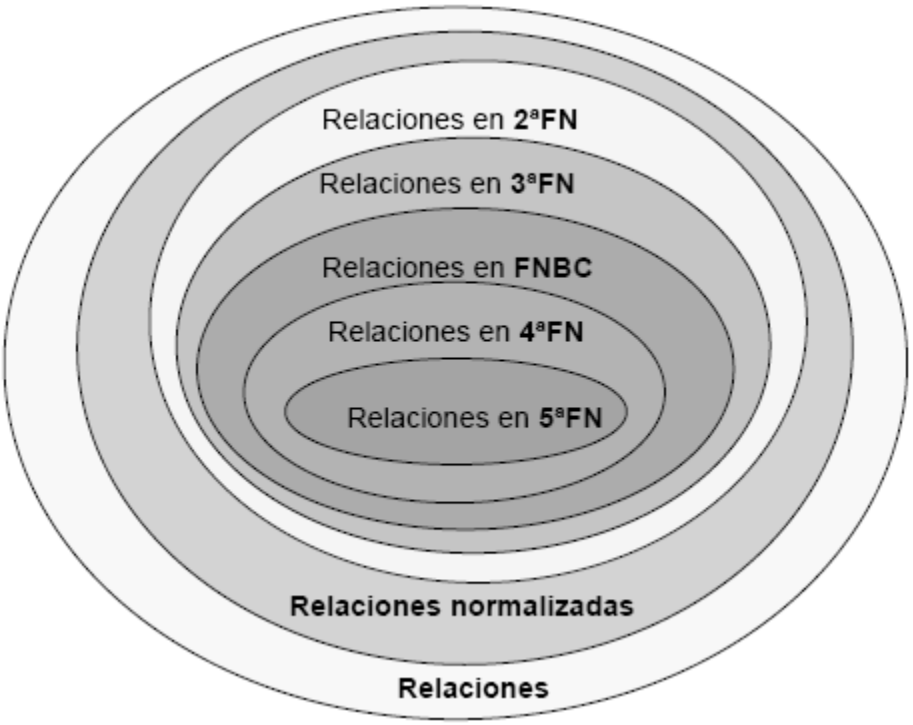


Figura 2. Fases de la normalización

Las tres primeras formas normales son suficientes para cubrir las necesidades de la mayoría de las bases de datos, las demás son opcionales. El creador de estas 3 primeras formas normales (o reglas) fue Edgar F. Codd. Éste introdujo la normalización en un artículo llamado A Relational Model of Data for Large Shared Data Banks Communications of the ACM, Vol. 13, No. 6, June 1970, pp. 377-387.

3. Primera forma normal (1FN)

Una tabla se encuentra en primera forma normal si impide que un atributo de una tupla pueda tomar más de un valor.

Se eliminan los grupos repetitivos. Los atributos han de ser atómicos, es decir, **cada atributo de la tabla toma un único valor del dominio correspondiente**. Gráficamente las celdas de la tabla contienen solo un valor, **en cada uno de los atributos sólo se puede incluir un dato, aunque sea compuesto, pero no se puede incluir una lista de datos**. Se trata de que cada atributo guarde la menor cantidad de información posible.

Para eliminar los grupos repetitivos puede ponerse cada a uno de ellos en una tabla aparte. Esa nueva tabla hereda la clave primaria de la relación en la que se encontraban.

TRABAJADOR		
DNI	**Nombre**	**Departamento**
12121212A	Andrés	Mantenimiento
12345345G	Andrea	Dirección Gestión

Visualmente es un tabla, pero no una tabla relacional (lo que en terminología de bases de datos relacionales se llama **relación**). No cumple la primera forma normal. Lo cumpliría si:

TRABAJADOR		
DNI	**Nombre**	**Departamento**
12121212A	Andrés	Mantenimiento
12345345G	Andrea	Dirección
12345345G	Andrea	Gestión

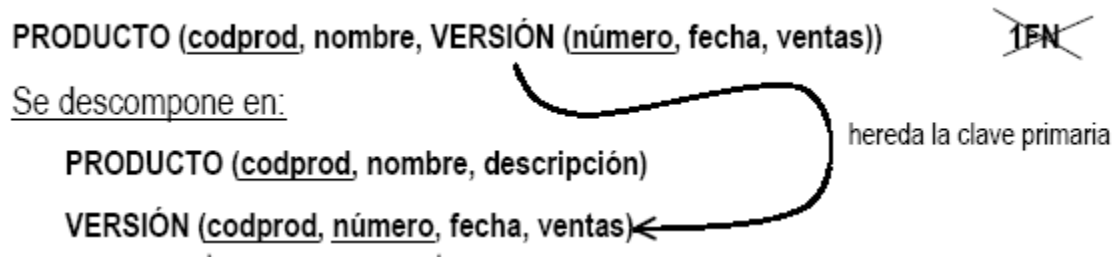
Esa tabla sí está en primera forma normal.

Para pasar una tabla a 1ª forma normal tenemos que identificar las columnas que tienen varios valores para uno concreto de la clave (grupos repetitivos). Hecho esto formaremos con ellas una nueva tabla eliminándolos de la original. La clave principal de la nueva tabla o relación estará formada por la concatenación de una o varias de sus columnas con la clave principal de la antigua.

PRODUCTO

codprod	nombre	VERSIÓN		
		número	fecha	ventas
LH4	Ladrillo hueco	1	1/3/1996	30.000
		2	1/8/1998	50.000
		3	1/2/2000	13.000
LP7	Ladrillo perforado	1	1/6/1996	70.000
		2	1/12/2000	

grupos repetitivos
(valores no atómicos)



Si el modelo E/R es correcto, todas las tablas están en 1ª forma normal.

4. Dependencias funcionales

4.1. Dependencia funcional

Se dice que un conjunto de atributos (**Y**) depende funcionalmente de otro conjunto de atributos (**X**) si para cada valor de **X** hay un único valor posible para **Y**. Simbólicamente se denota por **X→Y**.

Por ejemplo el nombre de una persona depende funcionalmente del DNI, para un DNI concreto sólo hay un nombre posible.

Al conjunto **X** del que depende funcionalmente el conjunto **Y** se le llama **determinante**. Al conjunto **Y** se le llama **implicado**.

Por ejemplo si conocemos el valor de **FechaDeNacimiento** podemos conocer el valor de **Edad.**

FechaDeNacimiento → Edad

Aquí al atributo **FechaDeNacimiento** se le conoce como un determinante, pues para cada valor de FechaDeNacimiento hay un valor único valor de Edad asociado.

Se puede leer de dos formas: *FechaDeNacimiento determina a Edad o Edad es funcionalmente dependiente de FechaDeNacimiento.*

4.2. Dependencia funcional completa

Un conjunto de atributos (**Y**) tiene una dependencia funcional completa sobre otro conjunto de atributos (**X**) si **Y** tiene dependencia funcional de **X** y además no se puede obtener de **X** un conjunto de atributos más pequeño que consiga una dependencia funcional de **Y**.

Por ejemplo en una tabla de clientes, el conjunto de atributos formado por el **nombre** y el **dni** producen una dependencia funcional sobre el atributo **apellidos**. Pero no es plena ya que el **dni** sólo también produce una dependencia funcional sobre **apellidos**.

El **dni** sí produce una dependencia funcional completa sobre el campo **apellidos**.

Una dependencia funcional completa se denota como **X⇒Y**

Se dice que el atributo (**Y**) es completamente dependiente de (**X**) si depende funcionalmente de (**X**) y no depende de ningún subconjunto de (**X**).

(artículo, cliente) ⇒ *cantidad* es una DF completa, pero

(artículo, cliente) ⇒ *precio* no es una DF completa puesto que *artículo* ⇒ *precio;*

4.3. Dependencia funcional transitiva

Es más compleja de explicar, pero tiene también utilidad. Se produce cuando tenemos tres conjuntos de atributos **X**, **Y** y **Z**. **Y** depende funcionalmente de **X** (X→Y), **Z** depende funcionalmente de **Y** (Y→Z). Además **X** no depende funcionalmente de **Y**. Entonces ocurre que **X** produce una dependencia funcional transitiva sobre **Z**. Esto se denota como:

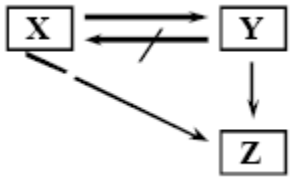
(X— →Z)

Por ejemplo si **X** es el atributo **Número de Clase** de un instituto, e **Y** es el atributo **Código Tutor**. Entonces **X→Y** (el tutor depende funcionalmente del número de clase).

Si **Z** representa el **Código del departamento**, entonces **Y→Z** (el código del departamento depende funcionalmente del código tutor, cada tutor sólo puede estar en un departamento).

Como no ocurre que **Y→X** (el código de la clase no depende funcionalmente del código tutor, puesto que un tutor lo puede ser de varias clases).

Entonces **X—→Z** (el código del departamento depende transitivamente del código de la clase).



Supongamos, por ejemplo, que en una relación los estudiantes solo pueden estar matriculados en un solo curso y supongamos que los cursos sólo pueden ser impartidos por un profesor.

ID_Estudiante → Curso
Curso → Profesor_Asignado
Curso ↛ ID_Estudiante

Entonces

ID_Estudiante —→ Profesor_Asignado

Entonces tenemos que **ID_Estudiante** determina a **Curso** y el **Curso** determina a **Profesor_Asignado**; indirectamente podemos saber a través del **ID_estudiante** el **Profesor_Asignado**. Entonces tenemos una dependencia transitiva.

De la normalización (lógica) a la implementación (física o real) puede ser sugerible tener en cuenta éstas dependencias funcionales para lograr mayor eficiencia en las tablas construidas

5. Segunda forma normal (2FN)

Dependencia completa. Una tabla **está en 2FN si y sólo si está en 1FN y si sus atributos no principales (que no pertenecen a la clave primaria) dependen de forma completa de la clave primaria (de todos los atributos de la clave primaria)**.

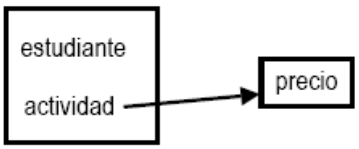
Se aplica en tablas con claves primarias compuestas por varios atributos, por tanto, toda tabla que tenga como clave primaria sólo un atributo está en 2FN si ya lo estaba en 1FN.

Si una relación R no está en 2ª FN, se puede normalizar descomponiendo esa relación en:

- Una relación con los atributos de clave primaria, más los atributos con dependencia completa de ella.
- Una relación para cada "parte" de la clave primaria, más los atributos que dependan funcionalmente de esa parte.

INSCRIPCIÓN (estudiante, actividad, precio) ~~2FN~~

actividad → precio



estudiante	actividad	precio
100	Tenis	1500
100	Yoga	1500
200	Tenis	1500
300	Escalada	5000

misma actividad, mismo precio.

Se descompone en

INSCRIPCIÓN (estudiante, actividad)

ACTIVIDAD (actividad, precio)

ALUMNOS				
DNI	**Cod Curso**	**Nombre**	**Apellido1**	**Nota**
12121219A	34	Pedro	Valiente	9
12121219A	25	Pedro	Valiente	8
3457775G	34	Ana	Fernández	6
5674378J	25	Sara	Crespo	7
5674378J	34	Sara	Crespo	6

Suponiendo que el **DNI** y el **número de curso** formen una clave principal para esta tabla, sólo la **nota** tiene dependencia funcional completa. El **nombre** y los **apellidos** dependen de forma completa del **DNI**. La tabla no está en 2FN. Para arreglarlo:

ALUMNOS		
DNI	**Nombre**	**Apellido1**
12121219A	Pedro	Valiente
3457775G	Ana	Fernández
5674378J	Sara	Crespo

ASISTENCIA		
DNI	**Cod Curso**	**Nota**
12121219A	34	9
12121219A	25	8
3457775G	34	6
5674378J	25	7
5674378J	34	6

6. Tercera forma normal (3FN)

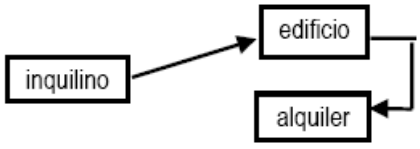
Se eliminan las dependencias transitivas. Una tabla **está en 3FN si y sólo si está en 2FN y todo atributo que no está en la clave primaria no depende transitivamente de la clave primaria**. El valor de esta columna debe depender directamente de la clave. **Todos los valores de una tabla deben identificarse únicamente por la clave directamente, y no por un campo intermedio no principal de la tabla que a su vez depende funcionalmente de la clave principal (dependencia transitiva)**.

Por lo tanto, podemos decir que una tabla no esta en 3FN cuando algún atributo depende funcionalmente de atributos que no son clave ni podrían haberlo sido puesto que se han quedado como claves alternativas

Para pasar una relación de 2FN a 3FN hay que eliminar las dependencias transitivas. Para ello, se eliminan de la tabla los atributos que dependen transitivamente y se ponen en una nueva relación con una copia de el atributo o atributos no clave de los que dependen.

INQUILINO (inqulino, edificio, alquiler) ~~3FN~~

edificio → alquiler



inquilino	edificio	alquiler
100	E04	50.000
200	E13	50.000
300	E09	65.000
400	E04	50.000

mismo edificio, mismo alquiler.

Se descompone en las tablas:

INQUILINO (inquilino, edificio)

EDIFICIO (edificio, alquiler)

ALUMNOS				
DNI	**Nombre**	**Apellido1**	**Cod Provincia**	**Provincia**
12121349A	Salvador	Velasco	34	Palencia
12121219A	Pedro	Valiente	34	Palencia
3457775G	Ana	Fernández	47	Valladolid
5674378J	Sara	Crespo	47	Valladolid
34568658S	Marina	Serrat	08	Barcelona

La **Provincia** depende funcionalmente del **código de provincia,** lo que hace que no esté en 3FN. El arreglo sería:

ALUMNOS			
DNI	**Nombre**	**Apellido1**	**Cod Provincia**
12121349A	Salvador	Velasco	34
12121219A	Pedro	Valiente	34
3457775G	Ana	Fernández	47
5674378J	Sara	Crespo	47
34568658S	Marina	Serrat	08

PROVINCIA	
Cod Provincia	**Provincia**
34	Palencia
47	Valladolid
08	Barcelona

7. Ejercicios

- Pasar a 3FN la siguiente tabla:

estudiante	nombre	apellido	DNI	dirección	codbeca	nombeca	requisito	fecha
0123	Carlos	Gil	159357	C/ Paz, 23	A223	EEUU	Ing. Sup.	10/10/98
7636	Paula	Tena	913752	C/ Río Po, 1	B567	ERASMUS	Ing. Téc.	12/11/98
7636	Paula	Tena	913752	C/ Río Po, 1	A223	EEUU	Ing. Sup.	14/10/98
7636	Paula	Tena	913752	C/ Río Po, 1	G654	DRAC	Ing. Sup.	15/09/99
0123	Carlos	Gil	159357	C/ Paz, 23	G654	DRAC	Ing. Sup.	17/09/98
9516	Andrés	Calpe	682432	Plz. Sol, 40	G654	DRAC	Ing. Sup.	12/09/99
0123	Carlos	Gil	159357	C/ Paz, 23	B567	ERASMUS	Ing. Téc.	12/11/98
9516	Andrés	Calpe	682432	Plz. Sol, 40	B567	ERASMUS	Ing. Téc.	23/11/99
0123	Carlos	Gil	159357	C/ Paz, 23	A223	EEUU	Ing. Sup.	12/10/99
3361	Lucía	Porcar	243115	Plz. Sol, 26	A223	EEUU	Ing. Sup.	12/10/99
...	...	...	...	...	...	...	...	...

SOLICITUD (<u>estudiante, codbeca, fecha</u>, nombre, apellido, DNI, dirección, nombeca, requisito)

- Pasar a 3FN la siguiente tabla:

PEDIDOS - LIBROS

IdPed	Fecha	*IdProv*	NomProv	DirecProv	*IdLib*	Precio	Cant	ISBN	Título	Lugar	Editor	Año	Pág	*IdAut*	Nombre
1	14/09/97	*1*	DiegoMarín	Merced, 32. 30043 - Murcia.	*1*	3200	2	84-7897-233-1	Diseño de BD	Madrid	Paraninfo	1996	225	*1*	Ernesto Rivero
2	1/10/97	*2*	Blackwells	16 The Avenue Oxford	*2*	1800	1	0-82112-462-3	Química Orgánica	Madrid	RA-MA	1996	310	*2*	Vicente Rodríguez
					3	2000	4	0-84131-460-7	Alquimia	Madrid	RA-MA	1995	522	*2*	Vicente Rodríguez
					4	3299	20	0-69213-517-8	Sistemas expertos	Murcia	U.Murcia	1997	257	*1*	Ernesto Rivero
					6	3200	2	0-71143-526-6	Bibliografía	Granada	U.Granada	1993	794	*3*	Carlos Rodero
3	4/11/97	*3*	Díaz de Santos	Magallanes, 25 28015 - Madrid.	*5*	4299	2	84-3112-462-1	Informática Documental	Oxford	Blackwells	1996	125	*3*	Carlos Rodero
					6	3800	3	0-71143-526-6	Bibliografía	Granada	U.Granada	1993	794	*3*	Carlos Rodero
					7	3250	1	0-63322-891-7	Teoría de gestión	Barcelona	Paraninfo	1995	173	*1*	Ernesto Rivero
					2	1850	2	0-82112-462-3	Química Orgánica	Madrid	RA-MA	1996	310	*2*	Vicente Rodríguez
4	10/12/97	*2*	Blackwells	16 The Avenue Oxford	*2*	1900	2	0-82112-462-3	Química Orgánica	Madrid	RA-MA	1996	310	*2*	Vicente Rodríguez
					7	3400	4	0-63322-891-7	Teoría de gestión	Barcelona	Paraninfo	1995	173	*1*	Ernesto Rivero

Clave primaria: (**Idped**)

PEDIDOS-LIBROS(IdPed, Fecha, IdProv, NomProv, DirecProv, IdLib, Precio, Cant, ISBN, Título, Lugar, Editor, Año, Pág, IdAot, Nombre)