



Normalización

EJERCICIO 1 - NORMALIZAR HASTA LA 3FN

Normalizar la siguiente tabla hasta la 3FN

ALUMNOS

DNI	NOMBRE	APELLIDO	C POSTAL	PROVINCIA	ASIGNAT	PROFESOR	NOTA
12345678	Silvia	Thomas	18019	Granada	Informática	A. Bonillo	9
12345678	Silvia	Thomas	18019	Granada	Matemáticas	N. Jáimez	7
34567890	Miguel A.	Pérez	14001	Córdoba	Matemáticas	N. Jáimez	8
34567890	Miguel A.	Pérez	14001	Córdoba	Lengua	S. López	9
23456789	Diego	Rodríguez	04720	Almería	Lengua	S. López	9
23456789	Diego	Rodríguez	04720	Almería	Informática	A. Bonillo	9

1FN

- ¿Está la relación ALUMNO en 1FN?

Podemos decir que la relación **ALUMNO** está en 1FN ya que no hay ningún atributo que no sea atómico, es decir, en cada tupla de la relación todos los atributos toman un único valor.

2FN

- ¿Está en 2FN?

Para hacer esta comprobación tenemos que ver que todos los atributos que no forman parte de la clave primaria tienen dependencia funcional completa respecto de dicha clave. Empezamos antes de nada determinando cuál es la clave primaria de esta relación.

Está claro que para que cada tupla de la relación sea única, la clave primaria de la relación **ALUMNO** tiene que estar compuesta por los atributos **DNI** y **ASIGNAT**.

ALUMNO

DNI	NOMBRE	APELLIDO	C POSTAL	PROVINCIA	ASIGNAT	PROFESOR	NOTA
12345678	Silvia	Thomas	18019	Granada	Informática	A. Bonillo	9
12345678	Silvia	Thomas	18019	Granada	Matemáticas	N. Jáimez	7
34567890	Miguel A.	Pérez	14001	Córdoba	Matemáticas	N. Jáimez	8
34567890	Miguel A.	Pérez	14001	Córdoba	Lengua	S. López	9
23456789	Diego	Rodríguez	04720	Almería	Lengua	S. López	9
23456789	Diego	Rodríguez	04720	Almería	Informática	A. Bonillo	9

Vemos las dependencias que se observan en la relación:

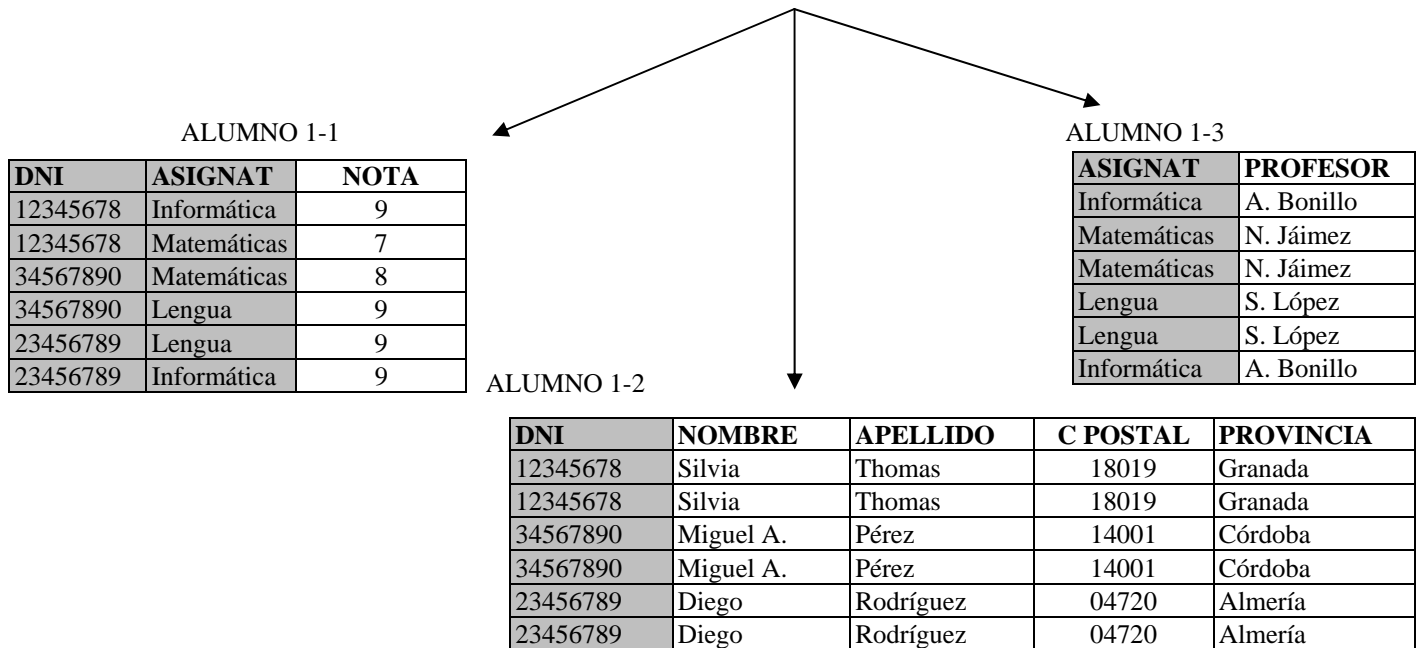
DNI --> NOMBRE, APELLIDO, C_POSTAL
C_POSTAL --> PROVINCIA
DNI, ASIGNAT --> NOTA
ASIGNAT --> PROFESOR

Luego **ALUMNO** no está en 2FN, ya que hay atributos que no dependen de toda la clave DNI, ASIGNAT sino que dependen de una parte de la clave . ¿Cómo lo solucionamos?



NOTA: Los atributos no claves que dependen de otros atributos no clave se mirarán en la 3FN

DNI	ASIGNAT	NOMBRE	APELLIDO	C POSTAL	PROVINCIA	PROFESOR	NOTA
12345678	Informática	Silvia	Thomas	18019	Granada	A. Bonillo	9
12345678	Matemáticas	Silvia	Thomas	18019	Granada	N. Jáimez	7
34567890	Matemáticas	Miguel A.	Pérez	14001	Córdoba	N. Jáimez	8
34567890	Lengua	Miguel A.	Pérez	14001	Córdoba	S. López	9
23456789	Lengua	Diego	Rodríguez	04720	Almería	S. López	9
23456789	Informática	Diego	Rodríguez	04720	Almería	A. Bonillo	9



Toda la información del alumno, solo depende del DNI, no del DNI y la ASIGNAT ==> no 2FN

La nota si depende del DNI y del ASIGNAT

PROFESOR solo depende de una parte de la clave ASIGNAT

Descomponiendo esta tabla en 3 tenemos:

ALUMNO 1-1 (**DNI**, **ASIGNAT**, NOTA)

ALUMNO 1-2 (**DNI**, NOMBRE, APELLIDO, CPOSTAL, PROVINCIA)

ALUMNO 1-3 (**ASIGNAT**, PROFESOR)

Ahora sí que todas las relaciones **están en 2FN**.

3FN

- ¿Están las relaciones obtenidas en 3FN?

Para comprobar si una relación está en 3FN tenemos que asegurarnos de que no existen dependencias funcionales transitivas en dicha relación, es decir si atributos no claves dependen de otros atributos no claves.

ALUMNO 1-1 si está en 3FN

ALUMNO 1-3 si está en 3FN



ALUMNO 1-2 no está en 3FN, que el atributo no clave PROVINCIA depende de otro atributo no clave C POSTAL

ALUMNO 1-2

DNI	NOMBRE	APELLIDO	C POSTAL	PROVINCIA
12345678	Silvia	Thomas	18019	Granada
12345678	Silvia	Thomas	18019	Granada
34567890	Miguel A.	Pérez	14001	Córdoba
34567890	Miguel A.	Pérez	14001	Córdoba
23456789	Diego	Rodríguez	04720	Almería
23456789	Diego	Rodríguez	04720	Almería

ALUMNO 2-1

DNI	NOMBRE	APELLIDO	C POSTAL
12345678	Silvia	Thomas	18019
12345678	Silvia	Thomas	18019
34567890	Miguel A.	Pérez	14001
34567890	Miguel A.	Pérez	14001
23456789	Diego	Rodríguez	04720
23456789	Diego	Rodríguez	04720

ALUMNO 2-2

C POSTAL	PROVINCIA
18019	Granada
18019	Granada
14001	Córdoba
14001	Córdoba
04720	Almería
04720	Almería

Al final tendríamos 4 tablas

ALUMNO 1-1 --> CALIFICACIÓN (DNI, ASIGNAT, NOTA)

ALUMNO 1-3 --> PROFESOR (ASIGNAT, PROFESOR)

ALUMNO 2-1 --> ALUMNO (DNI, NOMBRE, APELLIDO, CPOSTAL)

ALUMNO 2-2 --> CODIGO POSTAL (CPOSTAL, PROVINCIA)

CALIFICACION

DNI	ASIGNAT	NOTA
12345678	Informática	9
12345678	Matemáticas	7
34567890	Matemáticas	8
34567890	Lengua	9
23456789	Lengua	9
23456789	Informática	9

PROFESOR

ASIGNAT	PROFESOR
Informática	A. Bonillo
Matemáticas	N. Jáimez
Matemáticas	N. Jáimez
Lengua	S. López
Lengua	S. López
Informática	A. Bonillo

ALUMNO

DNI	NOMBRE	APELLIDO	C POSTAL
12345678	Silvia	Thomas	18019
12345678	Silvia	Thomas	18019
34567890	Miguel A.	Pérez	14001
34567890	Miguel A.	Pérez	14001
23456789	Diego	Rodríguez	04720
23456789	Diego	Rodríguez	04720

CODIGO POSTAL

C POSTAL	PROVINCIA
18019	Granada
18019	Granada
14001	Córdoba
14001	Córdoba
04720	Almería
04720	Almería



EJERCICIO 2 - NORMALIZAR HASTA LA 3FN

Para identificar la factura, hemos considerado como clave primaria el código de la factura y además, hemos deducido que necesariamente una factura debe poseer todos esos campos.

Este es el diseño inicial de las FACTURAS, al cual debemos aplicarle la metodología de las formas normales para ver si se trata de un buen diseño (aunque dados nuestros conocimientos previos podemos afirmar que no lo es). En una misma factura puede haber varios artículos

FACTURA

<p><i>Codigo_factura</i> Codigo_cliente Nombre_cliente Direccion_cliente Poblacion_cliente Fecha_factura Forma_pago Codigo_articulo_1 Descripcion_1 Cantidad_1 Importe_1 Tipo_IVA_1 ... Codigo_articulo_N Descripcion_N Cantidad_N Importe_N Tipo_IVA_N</p>
--



Primera forma normal (1FN)

Recordemos que una base de datos se considera que está en 1FN si cada atributo (campo) de una tabla contiene un solo valor atómico (simple). Un atributo que contiene varios valores puede derivar en una pérdida de datos y por lo tanto no nos interesa.

Analizando el diseño inicial de la tabla FACTURA, observamos la existencia de múltiples valores para los atributos siguientes: *Codigo_articulo*, *descripcion*, *cantidad*, *importe* e *IVA*. Por lo tanto, observamos que no cumple la condición de 1FN. La solución consiste en crear una nueva tabla, que podemos llamar DETALLE_FACTURA, a la cual se trasladan los datos repetitivos, en nuestro caso los datos referentes a los artículos (*codigo_articulo*, *descripcion*, *cantidad*, *importe* e *IVA*). Aplicando esto, el diseño de la base de datos para las facturas en 1FN sería:

FACTURA

<i>Codigo_factura</i>
<i>Codigo_cliente</i>
<i>Nombre_cliente</i>
<i>Direccion_cliente</i>
<i>Poblacion_cliente</i>
<i>Fecha_factura</i>
<i>Forma_pago</i>

DETALLE_FACTURA

<i>Codigo_factura</i>
<i>Codigo_articulo</i>
<i>Descripcion</i>
<i>Cantidad</i>
<i>Importe</i>
<i>Tipo_IVA</i>

Segunda forma normal (2FN)

La segunda forma normal, como la tercera que veremos a continuación, se relaciona con el concepto de dependencia funcional.

Entendemos como dependencia funcional a la relación que tienen los atributos (campos) de una tabla con otros atributos de la propia tabla. Un campo tiene dependencia funcional si necesita información de otro/s campo/s para poder contener un valor.

Una tabla se dice que esta en segunda forma normal (2FN) si sucede que:

- Está en 1FN
- Cada atributo (campo) no clave depende de la clave completa, no de parte de ella.

Por supuesto, una base de datos estará en 2FN si todas sus tablas lo están.

La idea intuitiva de la 2FN es identificar todas las tablas con una clave compuesta, pues todas las tablas con clave simple están por defecto en 2FN si están en 1FN, y comprobar que cada uno de los campos de esta tabla depende de la clave completa.

En nuestro ejemplo, la tabla FACTURA se encuentra en 2FN pues está en 1FN y su clave es simple. Sin embargo, la tabla DETALLE_FACTURA ha de ser analizada pues su clave es compuesta (esta formada por dos atributos).

Analizando la tabla DETALLE_FACTURA, observamos que el atributo *descripcion* depende únicamente del atributo *codigo_articulo* (la descripción de un artículo depende únicamente de qué artículo se trate y es completamente independiente de la factura), por lo cual la *descripcion* habrá de ser llevada a una nueva tabla junto con el atributo clave *codigo_articulo*.



Supongamos además que el cliente nos indica que en la factura, aunque cada artículo posee calculado su IVA, el tipo de IVA que aplica es común a toda la factura y no depende en cada factura de los artículos. En este caso, el atributo Tipo_IVA solo dependerá funcionalmente del codigo_factura y no depende de codigo_articulo, por lo cual ha de ser devuelto a la tabla FACTURA como un único atributo Tipo_IVA que depende solo de la clave de FACTURA (codigo_factura). Con estas consideraciones, el diseño de la base de datos para las facturas de la empresa expresado en 2FN sería:



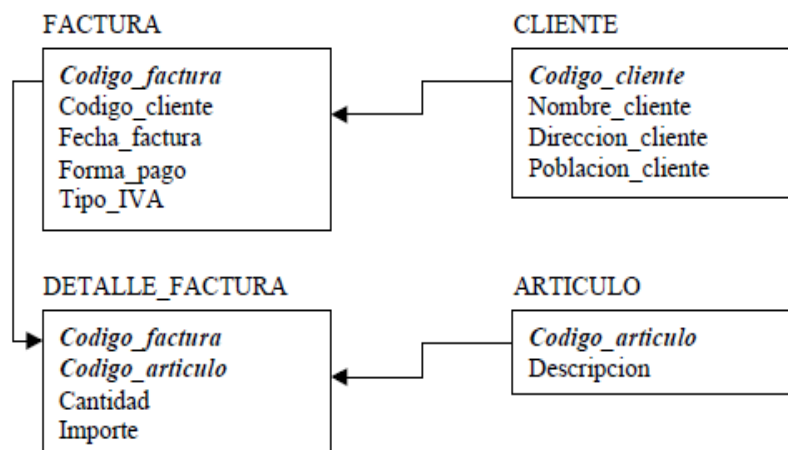
Tercera forma normal (3FN)

Recordemos que una tabla se dice que está en tercera forma normal (3FN) si:

- Está en 2FN.
- Todos los atributos que no son claves deben ser mutuamente independientes, es decir, un atributo no debe depender de otro atributo no clave de su tabla.

Si un atributo que no es clave depende de otro atributo que no es clave, la tabla posiblemente contiene datos acerca de más de una entidad, contradiciendo el principio de que cada tabla almacene información de una entidad.

En nuestro ejemplo, podemos observar que las tablas ARTICULO y DETALLE_FACTURA se encuentran en 3FN. Sin embargo, la tabla FACTURA no está en 3FN, pues los atributos *Nombre_cliente*, *Direccion_cliente* y *Poblacion_cliente* dependen funcionalmente del atributo *Codigo_cliente*, campo que no es clave. Por ello, debemos extraer estos atributos de la tabla FACTURA e incluirlos en una nueva tabla que haga referencia al cliente, tabla que llamaremos CLIENTE y que contendrá como clave primaria el *Codigo_cliente* y como atributos el *Nombre_cliente*, *Direccion_cliente* y *Poblacion_cliente*. Aplicando esto, nuestro diseño de la base de datos para las facturas da lugar a las tablas que pueden verse en la siguiente figura y que ya están en 3FN por lo que podemos considerar que es un buen diseño.





EJERCICIO 3 - NORMALIZAR HASTA LA 3FN

Tenemos la tabla de préstamos de libros

<u>CodLibro</u>	Titulo	Autor	Editorial	CodLector	Apel1	Apel2	Nombre	FechaDev
1001	Variable compleja	Murray Spiegel	McGraw Hill	501	Pérez	Gómez	Juan	15/04/2005
1004	Visual Basic 5	E. Petroustsos	Anaya	502	Ríos	Terán	Ana	17/04/2005
1005	Estadística	Murray Spiegel	McGraw Hill	503	Roca		René	16/04/2005
1006	Oracle University	Nancy Greenberg	Oracle Corp.	504	García	Roque	Luis	20/04/2005
1007	Clipper 5.01	Ramalho	McGraw Hill	501	Pérez	Gómez	Juan	18/04/2005

1FN

La tabla está en 1FN, al ser todos los atributos atómicos

2FN

La Segunda Forma Normal (2NF) pide que no existan dependencias parciales o dicho de otra manera, todos los atributos no clave deben depender por completo de la clave primaria. Actualmente en nuestra tabla tenemos varias dependencias parciales si consideramos como atributo clave el código del libro

El título es completamente identificado por el código del libro, pero el nombre del lector en realidad no tiene dependencia de este código, por tanto estos datos deben ser trasladados a otra tabla

<u>CodLibro</u>	Titulo	Autor	Editorial	CodLector	Apel1	Apel2	Nombre	FechaDev
1001	Variable compleja	Murray Spiegel	McGraw Hill	501	Pérez	Gómez	Juan	15/04/2005
1004	Visual Basic 5	E. Petroustsos	Anaya	502	Ríos	Terán	Ana	17/04/2005
1005	Estadística	Murray Spiegel	McGraw Hill	503	Roca		René	16/04/2005
1006	Oracle University	Nancy Greenberg	Oracle Corp.	504	García	Roque	Luis	20/04/2005
1007	Clipper 5.01	Ramalho	McGraw Hill	501	Pérez	Gómez	Juan	18/04/2005



LIBRO 1-1

CodLibro	Titulo	Autor	Editorial
1001	Variable compleja	Murray Spiegel	McGraw Hill
1004	Visual Basic 5	E. Petroustos	Anaya
1005	Estadística	Murray Spiegel	McGraw Hill
1006	Oracle University	Nancy Greenberg	Oracle Corp.
1007	Clipper 5.01	Ramalho	McGraw Hill

LIBRO 1-2

CodLector	Apel1	Apel2	Nombre
501	Pérez	Gómez	Juan
502	Ríos	Terán	Ana
503	Roca		René
504	García	Roque	Luis

LIBRO 1-3

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

3FN

Las tablas así están en 3FN

FNBC

Se podría optimizar todavía mas la tabla LIBRO 1-1 usando la FN Boyce-Codd que dice que cada atributo es clave candidata, entonces



LIBRO 2-1

CodLibro	Título	Autor	Editorial
1001	Variable compleja	Murray Spiegel	McGraw Hill
1004	Visual Basic 5	E. Petroustsos	Anaya
1005	Estadística	Murray Spiegel	McGraw Hill
1006	Oracle University	Nancy Greenberg	Oracle Corp.
1007	Clipper 5.01	Ramalho	McGraw Hill

LIBRO 2-2

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