

EJEMPLOS NORMALIZACIÓN

Ejemplo de diseño relacional inadecuado

Consideramos el diseño relacional de la tabla 1 para registrar la información de los profesores con los alumnos de cada uno y la calificación que han obtenido en los diversos créditos.

TABLA 1

DniProfesor	NombreProfesor	DniAlumno	NombreAlumno	Edad	Creditos	Nota
33333333	Juan	77777777	Ana	20	ADBD	4.5
33333333	Juan	88888888	Miguel	19	ADBD	5.7
33333333	Juan	77777777	Ana	20	SGBD	6
33333333	Juan	88888888	Miguel	19	SGBD	7
44444444	Maria	77777777	Ana	20	MET	6
44444444	Maria	88888888	Miguel	19	MET	5
44444444	Maria	77777777	Ana	20	LLC	4
44444444	Maria	88888888	Miguel	19	LLC	3

Verdad que convendréis que este diseño está pensado con los pies. Rápidamente, vemos los problemas siguientes:

- Hay información repetida, lo que puede provocar inconsistencias. Fijémonos que en caso de tener que modificar cualquiera de los valores de los campos que forman la clave primaria (DniProfesor, NombreProfesor, DniAlumno, NombreAlumno, Edad, Creditos), el cambio se efectuará en todas las filas en que aparece este valor.
- No puede haber valores nulos en las columnas que forman la clave primaria.
- Así, si no conocemos la edad de un alumno, tenemos un grave problema.
- En caso de llegar a la conclusión de que necesitamos almacenar más información de los profesores o los alumnos, habrá que añadir más columnas y repetir la información para cada fila en la que aparezca el profesor o alumno.
- Consultar la información en la tabla 1 puede convertirse pesado dada la gran cantidad de información diferente que contiene.

Hay diferentes grados de normalidad y, por tanto, de formas normales, las cuales cumplen la relación de inclusión de la figura 1, que debe interpretarse en el sentido que a medida que aumenta el nivel de la forma normal, la relación debe cumplir un conjunto de condiciones más restrictivo y, por tanto, continúa verificando las condiciones de las formas normales de nivel inferior.

FIGURA 1



Relación universal

Así, imaginamos que se quiere diseñar una base de datos para el control de los pedidos de compra de una organización determinada. Imaginemos que queremos incluir los conceptos correspondientes a número y fecha del pedido; código, descripción, cantidad y precio pactado por cada artículo solicitado; fecha prevista de entrega de el pedido; nombre (NomProv) y país (PaisProv) del proveedor; y moneda en la que se pacta el pedido. La relación universal se representa en la tabla 2.

TABLA 2. Relación universal para un esquema relacional ideado para una gestión de pedidos de compra.

NumPedido	FechaPedido	Articulo	Descripcion	Cantidad	Precio	FechaPrevista	NomProv	PaisProv	Moneda
22523	25-05-2000	PC3-500	PC Pentium III	5	150	1-06-2000	ARKANSAS	CHINA	EUR
22523	25-05-2000	PRO-15	Protector Pant.	5	8	1-06-2000	ARKANSAS	CHINA	EUR
22524	27-05-2000	PC3-500	PC Pentium III	15	145	5-06-2000	TOSCANA	ITALIA	USD
22524	27-05-2000	PRO-15	Protector Pant.	15	50	5-06-2000	TOSCANA	ITALIA	USD
22525	27-05-2000	INK430	Cartucho tinta	20	25	31-5-2000	ARKANSAS	CHINA	EUR

Verdad que hay mucha redundancia y poca organización? Evidentemente, el diseño relacional de una base de datos basado en la relación universal suele ser del todo incorrecto, y hace necesario aplicar un proceso de normalización con el fin de ir dividiendo la relación en otras relaciones de forma que alcancen grados de normalidad mejores, es decir, cumplan las restricciones correspondientes a las formas normales más elevadas.

Muy pocas veces se parte de la relación universal. La experiencia de los diseñadores provoca que, de entrada, ya se piense en relaciones que alcanzan un cierto grado de normalidad.

Dependencias funcionales

En la relación universal de la tabla 2 diríamos que, entre otros, la fecha de pedido depende funcionalmente del número de pedido, al igual que la fecha prevista, el nombre y el país del proveedor y la moneda. Podríamos escribir como sigue:

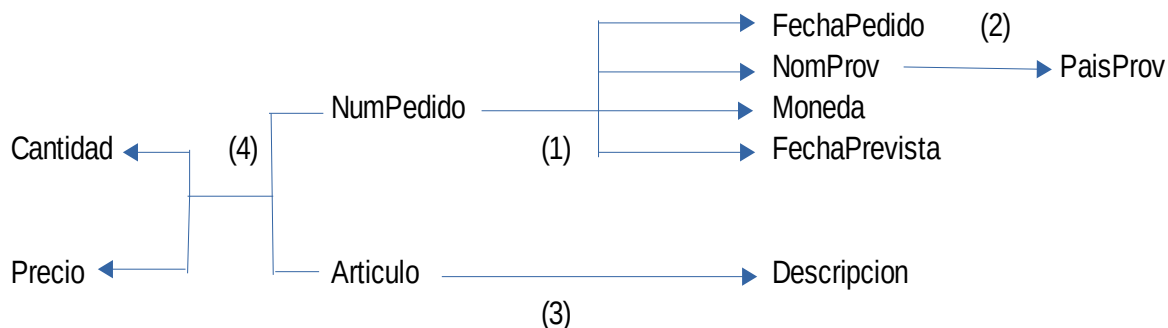
NumPedido -> FechaPedido
NumPedido -> FechaPrevista
NumPedido -> NomProv
NumPedido -> PaisProv
NumPedido -> Moneda

Al tratarse de diferentes atributos que dependen funcionalmente de un mismo atributo, escribiremos:

NumPedido -> FechaPedido, FechaPrevista, NomProv, PaisProv, Moneda

Es muy conveniente representar las dependencias funcionales de una relación mediante un esquema de dependencias funcionales. El esquema para la relación universal de la tabla 2 sería el que se muestra en la figura 2.

FIGURA 2. Ejemplo de esquema de dependencias funcionales



Fijémonos que hemos marcado los atributos que son clave de alguna de las entidades que forman parte de la relación: Artículo identifica el artículo y NumPedido identifica el pedido.

Fijémonos, también, que la pareja (NumPedido, Artículo) identifica la cantidad y precio de los artículos solicitados en el pedido.

En este esquema, se pueden ver las dependencias funcionales entre los atributos.

Se ve que FechaPedido, NomProv, Moneda y FechaPrevista dependen funcionalmente (1) de NumPedido, que PaisProv depende funcionalmente (2) de NomProv, y que Descripción depende funcionalmente (3) de Artículo. Asimismo, Cantidad y Precio dependen funcionalmente (4) de NumPedido y Artículo.

Es evidente que las dependencias (1), (2) y (3) son totales, ya que la parte izquierda de la dependencia (el implicador) está formada por un único atributo y, por tanto, es imposible que la parte derecha de la dependencia (el implicado) pueda depender de un subconjunto del implicador. La dependencia (4) también es total, ya que Cantidad y Precio dependen de la pareja (NumPedido, Artículo) y no de ningún subconjunto de ésta.

Una última apreciación sobre la dependencia funcional (4) de la figura 2: vemos que en un mismo pedido (NumPedido) no es posible tener varias veces el mismo artículo (Artículo), ya que los atributos Cantidad y Precio dependen funcionalmente de (NumPedido, Artículo). En caso de que fuera necesario tener varias veces el mismo artículo en un pedido, habría que utilizar algún otro atributo para identificar el artículo dentro de la pedido como, por ejemplo, el NumeroDeLinia del pedido.

En el caso anterior, vemos que NumPedido, Artículo, NomProv y la pareja (NumPedido, Artículo) son determinantes de la relación.

En el ejemplo de la figura 2, podemos decir que PaisProv depende transitivamente de NumPedido a través de NomProv.

Primera forma normal

Consideramos la relación universal de la tabla 3.

La relación de la tabla 3 no está en 1FN, ya que tiene atributos que pueden contener más de un valor. Vemos que este ejemplo incluye tres filas (pedidos 22.523, 22.524 y 22.525), y algunos atributos (Artículo, Descripcion, Cantidad y Precio) tienen varios valores para algunas de las filas.

TABLA 3.

NumPedido	FechaPedido	Articulo	Descripcion	Cantidad	Precio	FechaPrevista	NomProv	PaisProv	Moneda
22523	25-05-2000	PC3-500	PC Pentium III	5, 5	150, 8	1-06-2000	ARKANSAS	CHINA	EUR
		PRO-15	Protector Pant.						
22524	27-05-2000	PC3-500	PC Pentium III	15,15	145,5	5-06-2000	TOSCANA	ITALIA	USD
22525	27-05-2000	PRO-15	Protector Pant.	20	25	31-5-2000	ARKANSAS	CHINA	EUR

El proceso a seguir para lograr una 1FN es añadir tantas filas como sea necesario para cada uno de los diferentes valores del campo o campos que tengan valores no atómicos.

Así, en nuestro caso, obtenemos la relación en 1FN de la tabla 4.

TABLA 4.

NumPedido	FechaPedido	Articulo	Descripcion	Cantidad	Precio	FechaPrevista	NomProv	PaisProv	Moneda
22523	25-05-2000	PC3-500	PC Pentium III	5	150	1-06-2000	ARKANSAS	CHINA	EUR
22523	25-05-2000	PRO-15	Protector Pant.	5	8	1-06-2000	ARKANSAS	CHINA	EUR
22524	27-05-2000	PC3-500	PC Pentium III	15	145	5-06-2000	TOSCANA	ITALIA	USD
22524	27-05-2000	PRO-15	Protector Pant.	15	50	5-06-2000	TOSCANA	ITALIA	USD
22525	27-05-2000	INK430	Cartucho tinta	20	25	31-5-2000	ARKANSAS	CHINA	EUR

Segunda forma normal

Vemos que la clave primaria está formada por la pareja (NumPedido, Artículo) y que, en el esquema de dependencias funcionales asociado (figura 2), hay atributos fuera de la clave primaria que no tienen dependencia funcional completa de la clave.

En efecto, las dependencias funcionales (1), (2) y (3) nos presentan atributos que no tienen dependencia funcional total de la clave, formada por la pareja (NumPedido, Artículo).

El esquema de dependencias funcionales ayuda a ver las relaciones que han de aparecer. Así, en nuestro caso, de las dependencias (1), (3) y (4) obtenemos las relaciones en 2FN:

PEDIDO (NumPedido, FechaPedido, FechaPrevista, Moneda, NomProv, PaisProv)

ARTICULO (Articulo, Descripcion)

DETALLE (NumPedido, Articulo, Cantidad, Precio)

Es muy probable que este diseño fuera el propuesto como punto de partida, es decir: a menudo, al efectuar un diseño ya obtendremos relaciones que están en 2FN y, hasta y todo, en formas normales de nivel superior.

La información que hay en la tabla 4 correspondiente a la relación que acabamos de normalizar ahora pasa a estar repartida en tres tablas (tabla 5, tabla 6 y tabla 7).

TABLA 5. PEDIDO

NumPedido	FechaPedido	FechaPrevista	NomProv	PaisProv	Moneda
22523	25-05-2000	1-06-2000	ARKANSAS	CHINA	EUR
22524	27-05-2000	5-06-2000	TOSCANA	ITALIA	USD
22525	27-05-2000	31-5-2000	ARKANSAS	CHINA	EUR

TABLA 6. ARTICULO

Articulo	Descripcion
PC3-500	PC Pentium III
PRO-15	Protector Pant.
INK430	Cartucho tinta

TABLA 7. DETALLE

NumPedido	Articulo	Cantidad	Precio
22523	PC3-500	5	150
22523	PRO-15	5	8
22524	PC3-500	15	145
22524	PRO-15	15	50
22525	INK430	20	25

Por último, fijémonos que con el nuevo diseño se ha conseguido eliminar mucha redundancia y, por tanto, se reducen los problemas en las operaciones de actualización y consulta. Pero no desaparecen todos.

Tercera forma normal

Consideramos el diseño de las relaciones siguientes en 2FN ideadas para la gestión de los pedidos de compra de una organización:

PEDIDO (NumPedido, FechaPedido, FechaPrevista, Moneda, NomProv, PaisProv)

ARTICULO (Articulo, Descripcion)

DETALLE (NumPedido, Articulo, Cantidad, Precio)

Fijémonos que el país del proveedor aparece en cada pedido. Si partimos de la base de que el país donde reside el proveedor es único, ¿verdad que aún hay información redundante?

La tercera forma normal persigue la eliminación de los problemas motivados por la presencia de dependencias transitivas de los atributos que no forman parte de la clave primaria, respecto de la clave primaria.

Las relaciones ARTÍCULO y DETALLE que nos ocupan ya están en 3FN, pero consideramos la relación PEDIDO que contiene el país del proveedor.

El atributo PaisProv depende transitivamente de NumPedido a través de NomProv. Por lo tanto, esta relación no está en 3FN.

En nuestro caso, obtenemos las relaciones en 3FN:

PEDIDO (NumPedido, FechaPedido, FechaPrevista, Moneda, CodProv)

PROVEEDOR (CodProv, NomProv, PaisProv)

Fijémonos en que, al romper la relación inicial PEDIDO, ha parecido oportuno considerar un nuevo atributo (CodProv) que identifique mejor la nueva relación PROVEEDOR.

Este hecho no es imprescindible y no siempre será conveniente. podríamos haber considerado la rotura siguiente:

PEDIDO (NumPedido, FechaPedido, FechaPrevista, Moneda, NomProv)
PROVEEDOR (NomProv, PaisProv)

Ahora bien, para esta última posibilidad hemos escogido el nombre del proveedor como clave primaria de la nueva relación PROVEEDOR, y la experiencia nos aconseja definir un código que nos permita identificar de manera más clara que la que proporciona su nombre.

Teniendo en cuenta el diseño que incorpora el atributo CodProv, tendríamos la conversión de la tabla 5 en la tabla 8 y tabla 9.

TABLA 8. PEDIDO

NumPedido	FechaPedido	FechaPrevista	CodProv	Moneda
22523	25-05-2000	1-06-2000	ARK	EUR
22524	27-05-2000	5-06-2000	TOS	USD
22525	27-05-2000	31-5-2000	ARK	EUR

TABLA 9. PROVEEDOR

CodProv	NomProv	PaisProv
ARK	ARKANSAS	CHINA
TOS	TOSCANA	ITALIA

Forma normal de Boyce-Codd

Consideramos las naves de almacenamiento que hay en un gran mercado dedicadas guardar las mercancías de los vendedores del mercado. Imaginemos que cada nave guarda mercancía de un tipo concreto (carne fresca, pescado fresco, congelados, vegetales, bazar ...) y que cada vendedor puede depositar mercancía en diferentes naves según el tipo de mercancía de cada nave (un puesto de pescado del mercado puede dedicarse a vender pescado fresco y pescado congelado, por ejemplo). Ahora bien, toda la mercancía de unas mismas características de un vendedor se encuentra concentrada en una ubicación dentro una misma nave para minimizar al máximo los desplazamientos del vendedor.

Para tener constancia de qué tipo de material hay en cada nave, se diseña esta relación:

DEPOSITO (Vendedor, TipoMaterial, Nave, Ubicacion)

La tabla 10 nos ejemplifica la situación. Fijémonos que se encuentra en 3FN:

- Todos los atributos no pertenecientes a la llave (Nave y UBICACIÓN) tienen dependencia funcional total de la clave (2FN).

TABLA 10. DEPOSITO. Relación en 3FN.

Vendedor	TipoMaterial	Nave	Ubicacion
Jose	Pescado Fresco	15	S4
Maria	Pescado Fresco	25	S3
Ramón	Congelados	17	S2
Ana	Vegetales	20	S1
Ana	Pescado Fresco	25	S2
Maria	Bazar	10	S3

En efecto, la Nave y la Ubicacion dentro de la nave dependen del Vendedor y del TipoMaterial, ya que puede haber varias naves dedicadas a un tipo de material, pero todo el material similar de un vendedor se encuentra en una nave determinada. Al mismo tiempo, puede haber varias naves con material de un vendedor debido a la diferente tipología del material.

- Ningún atributo no perteneciente a la clave (Nave y Ubicacion) no depende transitivamente de la clave. En efecto, es imposible que haya ninguna dependencia transitiva de la clave, ya que no hay ningún atributo que pueda servir de puente para la transitividad. Pero esta relación, a pesar de estar en 3FN, presenta anomalías:

- Si en un momento dado una nave no tiene material de ningún vendedor, se pierde la información referente al tipo de mercancía que corresponde a la nave.

- Si cambia la descripción del tipo de mercancía asignada a una nave, hay que modificar tantas filas como vendedores con depósitos de aquel tipo de mercancía haya en la nave.

La anterior relación DEPOSITO no se encuentra en FNBC, ya que el atributo Nave es un determinante de la relación y TipoMaterial tiene dependencia funcional total de Nave, y en cambio Nave no es clave candidata.

Es decir, se da esta situación:

Nave -> TipoMaterial

Se verifica que toda relación en FNBC está en 3FN pero no al revés, como hemos podido comprobar en nuestro caso.

En nuestro caso, apartaremos de la relación el atributo TipoMaterial y obtendremos las relaciones en FNBC:

DEPOSITO (Vendedor, Nave, Ubicacion)

NAVE_TIPO (Nave, TipoMaterial)

En las roturas efectuados sobre una relación no normalizada para alcanzar relaciones

2FN y 3FN, hay que efectuar la división de manera que se preserven la información y las dependencias funcionales, hecho siempre posible en el paso a 2FN y 3FN. En el paso a FNBC, también siempre es posible efectuar la división manteniendo la información, pero no siempre es posible el mantenimiento de las dependencias funcionales.

En nuestro caso, la relación inicial DEPOSITO contenía las dependencias funcionales siguientes:

Vendedor, TipoMaterial -> Nave, Ubicacion
Nave -> TipoMaterial

Y, en las relaciones finales DEPOSITO y NAVE_TIPO, se ha perdido la dependencia funcional que indicaba que la Nave dependía de la pareja Vendedor y TipoMaterial:

DEPOSITO: Vendedor -> Nave, Ubicacion
NAVE_TIPO: Nave -> TipoMaterial

La dependencia funcional se ha perdido porque el concepto de dependencia funcional en el modelo relacional se define únicamente entre atributos de una misma relación, y en este se debería poder definir entre atributos de relaciones diferentes.

De todos modos, el concepto de integridad referencial intenta superar esta limitación.

Debido a la pérdida de dependencias funcionales, que no siempre se produce, a menudo no se normaliza a FNBC y se trabaja con relaciones en 3FN.

Cuarta forma normal

Consideramos la relación ESTUDIANTE (tabla 11) diseñada para almacenar los diversos créditos que cursa y las diversas actividades deportivas que practica.

ESTUDIANTE (Dni, Creditos, Deporte)

Es decir, la relación estudiante recoge la posibilidad de que un estudiante curse varios créditos y practique diversas actividades deportivas.

TABLA 11. ESTUDIANTE. Relación FNBC con redundancia.

Dni	Creditos	Deporte
10.000.000	SGBD	Baloncesto
10.000.000	ADBD	Baloncesto
10.000.000	SGBD	Futbol
10.000.000	ADBD	Futbol
20.000.000	PEM	Natación
20.000.000	ADBD	Natación
20.000.000	SGBD	Natación
20.000.000	PEM	Esgrima
20.000.000	ADBD	Esgrima
20.000.000	SGBD	Esgrima
15.000.000	PEM	Natación
15.000.000	PEM	Baloncesto

Esta relación se encuentra en FNBC y, aún así, hay redundancia, provocada por un nuevo concepto: las dependencias multivalentes.

Cuando tiene lugar una dependencia multivalente $A \twoheadrightarrow B$, también existe la dependencia multivalente $A \twoheadrightarrow X - (A \cup B)$, donde X indica el conjunto de todos los atributos de la relación. Es decir, las dependencias multivalentes se presentan por parejas.

En nuestro caso (tabla 11) se verifica lo siguiente:

$Dni \twoheadrightarrow Creditos$

$Dni \twoheadrightarrow Deporte$

Para alcanzar la 4FN a partir de una relación $R(A, B, C)$ que tiene una dependencia multivalente $A \twoheadrightarrow B$, hay que descomponer la relación R en dos relaciones $R_1(A, B)$ y $R_2(A, C)$.

En este caso, obtenemos las relaciones siguientes (tabla 12 y tabla 13):

CREDITOS_EN_CURSO (Dni, Creditos)

DEPORTE_EN_PRACTICA (Dni, Deporte)

TABLA 12. CREDITOS_EN_CURSO. Relación en 4FN.

Dni	Creditos
10.000.000	SGBD
10.000.000	ADBD
20.000.000	PEM
20.000.000	ADBD
20.000.000	SGBD
15.000.000	PEM

TABLA 13. DEPORTE_EN_PRACTICA. Relación en 4FN.

Dni	Deporte
10.000.000	Baloncesto
10.000.000	Futbol
20.000.000	Natación
20.000.000	Esgrima
15.000.000	Natación
15.000.000	Baloncesto

A menudo, la descomposición causada por las dependencias multivalentes se efectúa antes de las descomposiciones para alcanzar los niveles 2FN, 3FN y FNBC. En esta situación, en las relaciones obtenidas, se aplicarán las comprobaciones para conseguir que estén en 2FN, 3FN y FNBC.

Quinta forma normal

Consideramos la relación PROFESOR (tabla 14) diseñada para gestionar los profesores de una determinada institución escolar que tiene diferentes centros de docencia.

Cada profesor está autorizado a impartir unas determinadas especialidades docentes que puede poner en práctica en cualquiera de los centros docentes de la institución escolar.

Asimismo, cada profesor puede ejercer, además de la docencia, diferentes tareas (Cargos, tutoría pedagógica, tutoría técnica ...) en varios centros de la institución escolar.

PROFESOR (CodigoProf, Centro, Especialidad, Tarea)

TABLA 14. Relación en FNBC con dependencias multivalentes

CodigoProf	Centro	Especialidad	Tarea
P1	C1	Matemáticas	Tutor
P1	C2	Matemáticas	Tutor
P1	C2	Informática	Aula Inf.
P2	C1	Valenciano	Coordinador
P2	C2	Castellano	Tutor

Esta relación es FNBC y se aprecia una especie de dependencia multivalente, la que no se puede solucionar por la vía de la descomposición. En efecto, es muy fácil pensar en una descomposición en las tres relaciones siguientes:

CENTRO_DE_PROFESOR (CodigoProf, Centro)

ESPECIALIDAD_DE_PROFESOR (CodigoProf, Especialidad)

TAREA_DE_PROFESOR (CodigoProf, Tarea)

Esta descomposición es errónea, ya que si aplicamos el natural-join de las tres relaciones no obtenemos la relación inicial sino que obtenemos una relación con muchas más instancias (tabla 15).

TABLA 15. Relación obtenida del "natural-join" de las relaciones.

CodigoProf	Centro	Especialidad	Tarea
P1	C1	Matemáticas	Tutor
P1	C1	Matemáticas	Aula Inf.
P1	C1	Informática	Tutor
P1	C1	Informática	Aula Inf.
P1	C2	Matemáticas	Tutor
P1	C2	Matemáticas	Aula Inf.
P1	C2	Informática	Tutor
P1	C2	Informática	Aula Inf.
P2	C1	Valenciano	Coordinador
P2	C1	Valenciano	Tutor
P2	C1	Castellano	Coordinador
P2	C1	Castellano	Tutor
P2	C2	Valenciano	Coordinador
P2	C2	Valenciano	Tutor
P2	C2	Castellano	Coordinador
P2	C2	Castellano	Tutor

Queda claro, pues, que el método utilizado en este caso no es correcto y esto es debido a que, en esta situación, hay lo que se llama dependencias mutuas entre los atributos de la relación. Las dependencias mutuas provocan que la descomposición de la relación en otras relaciones (proyecciones del original) no verifique que su natural-join coincide con la relación original.

Volvamos al ejemplo de descomposición anterior: la relación PROFESOR se ha descompuesto en tres relaciones, CENTRO_DE_PROFESOR, ESPECIALIDAD_DE_PROFESOR y TAREA_DE_PROFESOR, y hemos podido comprobar que la relación profesor no satisface una dependencia de reunión respecto a CENTRO_DE_PROFESOR, ESPECIALIDAD_DE_PROFESOR y TAREA_DE_PROFESOR.

Necesitamos encontrar una dependencia de reunión para la relación PROFESOR, es decir, encontrar una descomposición tal que su natural-join recupere la relación original.

Fijémonos en la descomposición siguiente:

PROFESOR (CodigoProf, Centro, Especialidad, Tarea)
PCE = PROFESOR [CodigoProf, Centro, Especialidad]
PCT = PROFESOR [CodigoProf, Centro, Tarea]
PET = PROFESOR [CodigoProf, Especialidad, Tarea]

Donde PCE, PCT y PET son secciones de la tabla original.

En esta situación, vemos que si efectuamos el natural-join de las tres relaciones PCE, PCT y PET obtenemos la relación original.

Por lo tanto, la relación PROFESOR de nuestro ejemplo no se encuentra en 5FN, ya que hemos encontrado la dependencia de reunión PROFESOR * (PCE, PCT, PET) en la que las relaciones PCE, PCT y PET no están constituidas por claves candidatas de PROFESOR.

Así, en nuestro ejemplo, la relación PROFESOR desaparecería para dar paso a las tres relaciones PCE, PCT y PET en que se basa la dependencia de reunión encontrada.

PCE (CodigoProf, Centro, Especialidad)

PCT (CodigoProf, Centro, Tarea)

PET (CodigoProf, Especialidad, Tarea)