

EJEMPLO DE NORMALIZACIÓN

A partir de estas dos tablas y sabiendo que:

- CODIGO y NOMBRE de PROYECTO son equivalentes,
- Un empleado participa en varios proyectos,
- Un departamento tiene un empleado como director,
- Un proyecto tiene un empleado como director.

Indica en qué FN se encuentran y normalizar hasta FNBC.

PROYECTO (CODIGO, NOMBRE, FECHA-I, FECHA-F, DNI-PARTICIPANTE, HORAS-PARTICIPANTE, DNI-DIRECTOR, PRESUPUESTO)

EMPLEADO (DNI, NOMBRE, TITULACIÓN, COD-DPTO, NOMBRE-DPTO, DNI-DIRECTOR-DPTO)

SOLUCIÓN:

PROYECTO: está en 1FN (Si la clave fuera sólo CODIGO diríamos que ni siquiera en 1FN porque entonces tendría atributos multivaluados ya que un proyecto tiene varios participantes).

No está en 2FN porque existen dependencias funcionales parciales:

CODIGO → NOMBRE, FECHA-I, FECHA-F, DNI-DIRECTOR, PRESUPUESTO

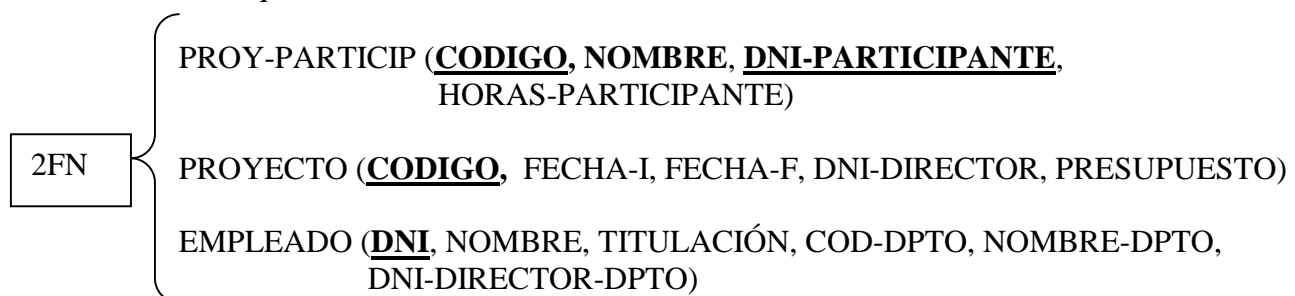
CODIGO, DNI-PARTICIPANTE → HORAS-PARTICIPANTE

NOMBRE, DNI-PARTICIPANTE → HORAS-PARTICIPANTE

Para que PROYECTO esté en 2FN hay que quitar los atributos **que no formen parte de ninguna clave** (que son FECHA-I, FECHA-F, DNI-DIRECTOR, PRESUPUESTO) y llevarlos a una tabla donde la clave sea CODIGO de proyecto. En la tabla original se quedan sólo los atributos que dependen de manera completa de la clave, y los que formen parte de alguna clave candidata también.

EMPLEADO está en 1FN si suponemos que solo tiene una titulación, y pertenece a un único departamento (que sólo tiene un nombre). También está en 2FN porque la clave no es compuesta, luego las dependencias funcionales son totales.

Cómo se quedan las tablas en 2FN:



Vamos a ver si están en 3FN: (Para todos los atributos que no formen parte de alguna clave, la dependencia funcional no puede ser transitiva)

PROY-PARTICIP: está en 3FN, no existen dependencias transitivas (ya hemos visto antes que NOMBRE forma parte de una clave y HORAS-PARTICIPANTE depende de la clave).

PROYECTO: está en 3FN, todos los atributos (FECHA-I, FECHA-F, DNI-DIRECTOR, PRESUPUESTO) dependen de CODIGO y no hay ninguno de estos atributos que dependa de otro que no sea CODIGO.

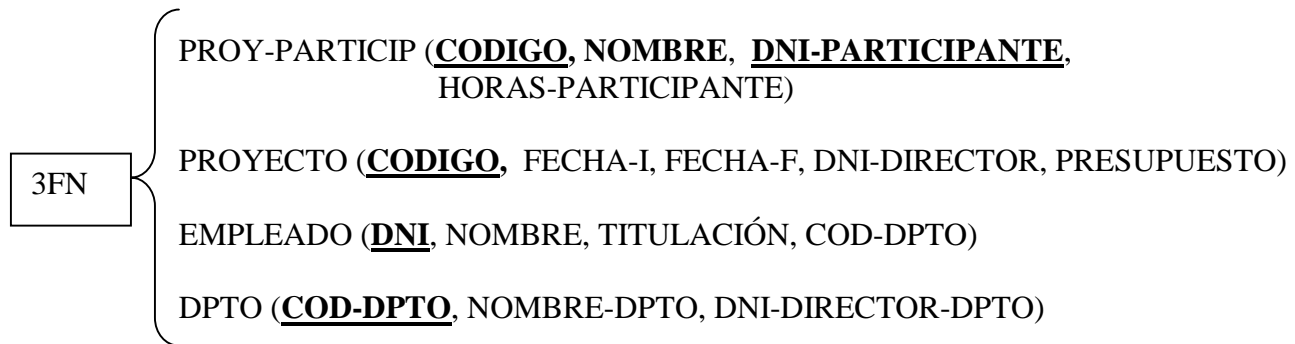
EMPLEADO: no está en 3FN, porque:

DNI → NOMBRE, TITULACIÓN, COD-DPTO, NOMBRE-DPTO, DNI-DIRECTOR-DPTO

Pero COD-DPTO → NOMBRE-DPTO, DNI-DIRECTOR-DPTO,

Por tanto estos dos atributos dependen de DNI de manera transitiva, y deben ir a una tabla donde la clave sea COD-DPTO.

Así quedarían en 3FN:



Vamos a ver ahora si están en FNBC (todo determinante debe ser clave, es decir todos aquellos atributos de los que dependan otros, tienen que ser clave, no vale que formen parte de una clave, tienen que ser clave)

Están en FNBC: PROYECTO, EMPLEADO y DPTO

PROY-PARTICIP no está en FNBC porque como ya vimos, se tienen las siguientes dependencias:

CODIGO, DNI-PARTICIPANTE → HORAS-PARTICIPANTE

NOMBRE, DNI-PARTICIPANTE → HORAS PARTICIPANTE

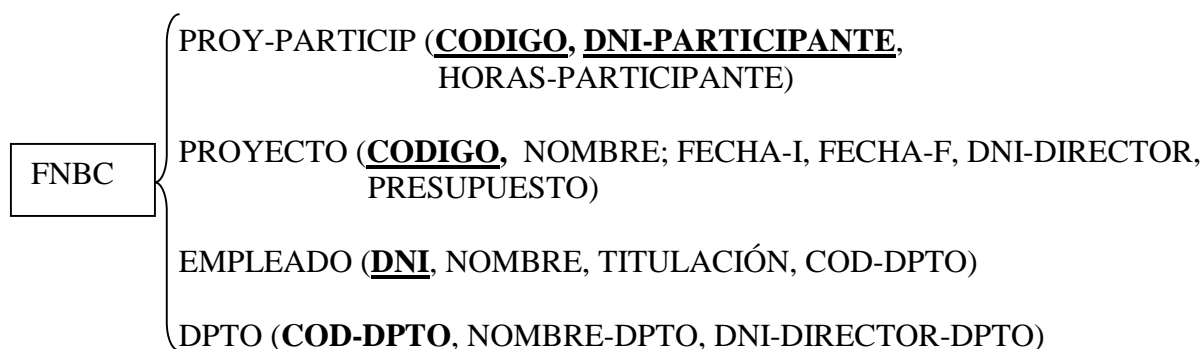
(De momento estos dos determinantes son clave (el primero es clave primaria y el otro clave alternativa))

Pero también se tiene que:

CODIGO → NOMBRE y NOMBRE → DNI, o sea, son equivalentes CODIGO ↔ NOMBRE y ni CODIGO ni NOMBRE (que son determinantes porque de ellos depende algún atributo) son clave.

Por tanto para que esta tabla esté en FNBC habrá que quitar de ella el atributo NOMBRE y llevarlo a una tabla donde la clave sea CODIGO.

De manera que las tablas quedan de esta manera en FNBC:



Ejemplo sencillo normalización

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 Díaz	Administrativo	1500	adiaz@ecn.es; ana32@gmail.com
...

Primera forma normal (1FN)

Una tabla está en 1FN si sus atributos contienen valores atómicos. En el ejemplo, podemos ver que el atributo *emails* puede contener más de un valor, por lo que viola 1FN.

En general, tenemos una relación R con clave primaria K . Si un atributo M viola la condición de 1FN, tenemos dos opciones.

Solución 1: duplicar los registros con valores repetidos

En general, esta solución pasa por sustituir R por una nueva relación modificada R' , en la cual:

- El atributo M que violaba 1FN se elimina.
- Se incluye un nuevo atributo M' que solo puede contener valores simples, de modo que si $R'[M']$ es uno de los valores que teníamos en $R[M]$, entonces $R'[K] = R[K]$. En otras palabras, para una tupla con n valores duplicados en M , en la nueva relación habrá n tuplas, que sólo varían en que cada una de ellas guarda uno de los valores que había en M .
- La clave primaria de R' es (K, M') , dado que podrá haber valores de K repetidos, para los valores multivaluados en M .

Siguiendo el ejemplo, tendríamos el siguiente esquema para la nueva tabla

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

EMPLEADOS' (a) con clave primaria (*nss*, *email*):

Solución 2: separar el atributo que viola 1FN en una tabla

En general, esta solución pasa por:

sustituir R por una nueva relación modificada R' que no contiene el atributo M .

Crear una nueva relación $N(K, M')$, es decir, una relación con una clave ajena K referenciando R' , junto al atributo M' , que es la variante mono-valuada del atributo M .

La nueva relación N tiene como clave (K, M') .

Siguiendo el ejemplo, tendríamos el siguiente esquema para la nueva tabla EMPLEADOS' (b)

nss	nombre	puesto	salario
111	Juan Pérez	Jefe de Área	3000
222	José Sánchez	Administrativo	1500
333	Ana Díaz	Administrativo	1500
...

Y además tendríamos una nueva tabla EMAILS con clave primaria $(nss, email)$:

nss	email
111	juanp@ecn.es
111	jefe2@ecn.es
222	jsanchez@ecn.es
333	adiaz@ecn.es
333	ana32@gmail.com
...	...

Segunda forma normal (2FN)

Un esquema está en 2FN si:

1º.-Está en 1FN.

2º.- Todos los atributos no principales (los que no formen parte de ninguna clave candidata) tienen dependencia funcional completa respecto de todas las claves candidatas existentes en el esquema.

La 2FN se aplica a las relaciones que tienen claves primarias compuestas por dos o más atributos. Si una relación está en 1FN y su clave primaria es simple (tiene un solo atributo), entonces también está en 2FN. Por tanto, de las soluciones anteriores, la tabla EMPLEADOS' (b) está en 1FN (y la tabla EMAILS no tiene atributos no clave), por lo que el esquema está en 2FN. Sin embargo, tenemos que examinar las dependencias funcionales de los atributos no clave de EMPLEADOS' (a). Las dependencias funcionales que tenemos son las siguientes:

$nss \rightarrow nombre, salario, email$

$puesto \rightarrow salario$

Como la clave es (nss, email), las dependencias de nombre, salario y email son *incompletas*, por lo que la relación no está en 2FN.

En general, tendremos que observar los atributos no clave que dependan de parte de la clave.

Para solucionar este problema, tenemos que hacer lo siguiente para los grupos de atributos con dependencia incompleta M :

Eliminar de R el atributo M .

Crear una nueva relación N con el atributo M y la parte de la clave primaria K de la que depende, que llamaremos K' .

La clave primaria de la nueva relación será K' .

Siguiendo el ejemplo anterior, crearíamos una nueva relación con los atributos que tienen dependencia incompleta:

nss	nombre	puesto	salario
111	Juan Pérez	Jefe de Área	3000
222	José Sánchez	Administrativo	1500
333	Ana Díaz	Administrativo	1500
...

Y al eliminar de la tabla original estos atributos nos quedaría:

nss	email
111	juanp@ecn.es
111	jefe2@ecn.es
222	jsanchez@ecn.es
333	adiaz@ecn.es
333	ana32@gmail.com
...	...

Como vemos, la solución a la que llegamos es la misma que en la otra opción de solución para el problema de 1FN.

Tercera forma normal (3FN)

Una relación está en tercera forma normal si, y sólo si:

1º.- está en 2FN

2º.- y, además, cada atributo no principal no depende transitivamente de la clave primaria.

Por lo tanto, a partir de un esquema en 2FN, tenemos que buscar dependencias funcionales entre atributos que no formen parte de ninguna clave candidata.

En general, tenemos que buscar dependencias transitivas de la clave, es decir, secuencias de dependencias como la siguiente: $K \rightarrow A$ y $A \rightarrow B$, donde A y B no pertenecen a la clave. La solución a este tipo de dependencias está en separar en una tabla adicional N el/los atributos B , y poner como clave primaria de N el atributo que define la transitividad A .

Siguiendo el ejemplo anterior, podemos detectar la siguiente transitividad:

`nss->puesto`

`puesto->salario`

Por lo tanto la descomposición sería la siguiente:

nss	nombre	puesto
111	Juan Pérez	Jefe de Área
222	José Sánchez	Administrativo
333	Ana Díaz	Administrativo
...

En la nueva tabla PUESTOS, la clave sería el puesto, que también queda como clave ajena referenciando la tabla EMPLEADOS. El resto de las tablas quedan como estaban.

puesto	salario
Jefe de Área	3000
Administrativo	1500
...	...