

UD 3-3. SOLUCIONES EJERCICIOS. DISEÑO LOGICO.

Base de Datos
CFGs DAW

PARTE 2

Francisco Aldarias Raya
paco.aldarias@ceedcv.es

2019/2020

Fecha 15/11/19

Versión:191115.1029

Licencia



Reconocimiento - NoComercial - CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:



Importante



Atención



Interesante

Revisiones.

15/11/2019. Ejercicio 1. Datos rellenos.

ÍNDICE DE CONTENIDO

1.EJERCICIO 1 localidad.....	3
2.EJERCICIO 2. proveedor.....	5
3.EJERCICIO 3. vino.....	9

UD03-3. SOLUCIONES EJERCICIOS. DISEÑO LOGICO. PARTE 2

El siguiente bloque de problemas son para practicar. No hay que entregarlos para evaluar. AL final de la semana se pondrán las soluciones. Si tienes dudas de tu solución comparte la en el foro.

1. EJERCICIO 1 LOCALIDAD

Se desea almacenar información de Provincias, Comarcas y Localidades de Valencia. Partimos del siguiente modelo lógico (modelo relacional) y se pretende normalizar hasta 3FN.

Sólo interesa guardar el nombre y el código tanto de la provincia, comarca y localidad.

Restricciones:

- Una localidad no puede estar en más de una comarca.
- Una comarca no puede estar en más de una provincia.

Diseño lógico:

Provincia(cod_prov, nom_prov)

CP: {cod_prov}

Comarca(cod_com, nom_com, cod_prov)

CP: {cod_com}

VNN: {cod_prov}

CAj: {cod_prov} --> Provincia

Localidad(cod_loc, nom_loc, cod_com)

CP: {cod_loc}

VNN: {cod_com}

CAj: {cod_com} --> Comarca

Solución:

1FN

Como ninguna de las relaciones anteriores tiene atributos multivaluados o compuestos, todas ellas están en 1FN.

2FN

Hay que eliminar las DF parciales. Como ninguna de las relaciones anteriores tiene claves compuestas, todas ellas están en 2FN.

3FN

Hay que eliminar las DF transitivas. Para ello realizamos un diagrama de DF para cada relación:

Provincia:

$\text{cod_pro} \rightarrow \text{nombre}$

Como no hay DF transitivas, la relación está en 3 FN.

Veamos la siguiente:

Comarca:

$\text{cod_om} \rightarrow \text{cod_prov}, \text{nombre}$

De la misma manera, comprobamos que esta relación tampoco tiene DF transitivas.

Por último, analicemos la relación Localidad:

Localidad:

$\text{cod_loc} \rightarrow \text{cod_com}, \text{nombre}$

Donde vemos, que de la misma manera, tampoco tiene DF transitivas, por lo que podemos concluir que las tres relaciones ya están en 3FN

Rellenando los datos se puede ver como quedaría:

PROVINCIA		
Cod_pro	Nombre	
Castellon	12	
Valencia	46	
Alicante	3	

COMARCA		
cod_comarca	nom_comarca	cod_pro
24	Vall d'Albaida	46
27	L,Alcoia	3
23	La costera	46
8	Alt Palancia	12

LOCALIDAD		
cod_localidad	nom_localidad	cod_comarca
46839	Guadasseguies	24
46830	Beniganim	24
3820	Cocentaina	27
12400	Segorve	8

2. EJERCICIO 2. PROVEEDOR

Normaliza a 3FN las siguientes relaciones:

Artículo(cod_art, descrip, color, peso)

CP: {cod_art}

Proveedor(cod_prov, cod_art, nombre, provincia, ciudad, cantidad)

CP: {cod_prov, cod_art}

CA: {cod_art} --> Artículo

Considera las siguientes dependencias funcionales (DF):

- cod_prov --> nombre, provincia, ciudad
- cod_art --> descripción, color, peso
- ciudad --> provincia
- cod_prov, cod_art --> cantidad

Solución:

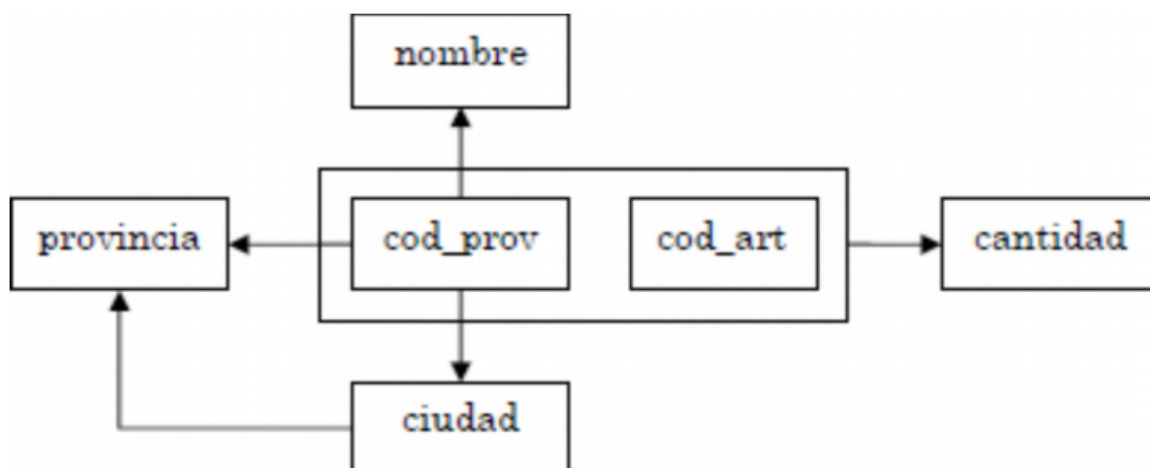
1FN

Hay que eliminar atributos compuestos, ni multi-evaluados, ni derivados. Como ninguna de las dos relaciones anteriores tiene atributos de ese tipo, ya están en 1FN.

2FN

Hay que eliminar las DF parciales. Como esto sólo afecta a las relaciones cuya clave primaria sea compuesta, podemos afirmar que la relación Artículo ya se encuentra en 2FN.

Para ver si Proveedor está en 2FN primero tenemos que obtener el diagrama de dependencias:



Se puede observar que hay atributos que dependen parcialmente de la clave.

Para pasar esta relación a 2FN haremos lo siguiente:

- En Proveedor dejaremos sólo la información que depende de cod_prov.
- Crearemos una nueva relación llamada Pedidos dónde figure la cantidad de artículos pedidos a un determinado proveedor.

Proveedor(cod_prov, nombre, provincia, ciudad)

CP: {cod_prov}

Pedidos(cod_prov, cod_art, cantidad)

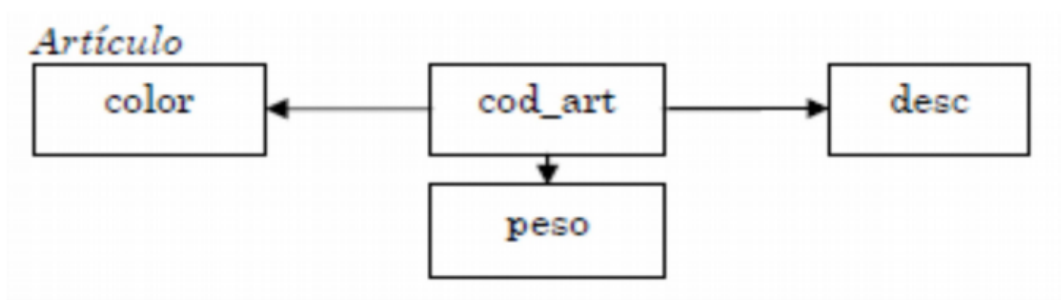
CP: {cod_prov, cod_art}

CAj: {cod_art} --> Artículo

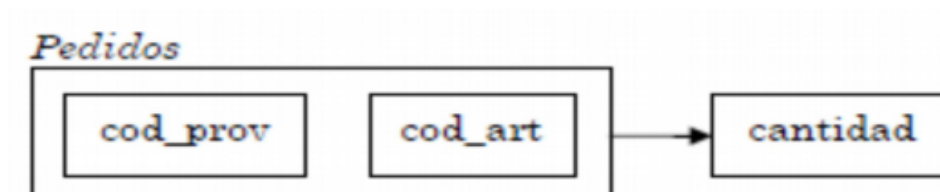
CAj: {cod_prov} --> Proveedor

3FN

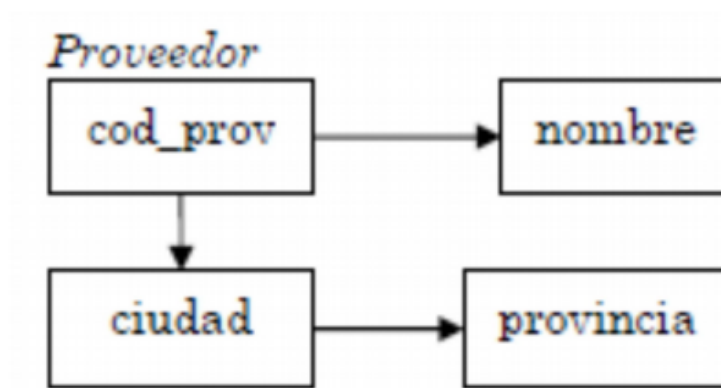
Hay que eliminar las DF transitivas. Para ello volvemos a realizar un diagrama de DF para cada relación:



Como no hay ninguna dependencia transitiva, la relación *Artículo* se encuentra en 3 FN.



Como no hay ninguna dependencia transitiva, la relación *Pedidos* se encuentra en 3 FN.



Como se puede observar aquí si que hay una dependencia transitiva. Para eliminarla creamos una nueva tabla llamada Ciudad que asocie ciudades con provincia y eliminamos el atributo provincia de la tabla Proveedor.

Ciudad(ciudad, provincia)

CP: {ciudad}

Proveedor(cod_prov, nombre, ciudad)

CP: {cod_prov}

CAj: {ciudad} --> Ciudad

Así pues el conjunto final de relaciones en 3FN es el siguiente:

Pedidos(cod_prov, cod_art, cantidad)

CP: {cod_prov, cod_art}

CAj: {cod_art} --> Artículo

CAj: {cod_prov} --> Proveedor

Ciudad(ciudad, provincia)

CP: {ciudad}

Proveedor(cod_prov, nombre, ciudad)

CP: {cod_prov}

CAj: {ciudad} --> Ciudad

3. EJERCICIO 3. VINO

Considera el siguiente esquema lógico:

Catador(dni_catador, nombre, experiencia, dni_conyuge, edad, fecha_nacimiento)

CP: {dni_catador}

edad = Año_Actual - fecha_nacimiento

Vino(nom_vino, añada, precio, bodega)

CP: {nom_vino}

Cata(dni_catador, nom_vino, nota_aromas, nota_sabor, tipo_vino)

CP: {dni_catador, nom_vino}

CAj: {dni_catador} --> Catador

CAj: {nom_vino} --> Vino

Composición(nom_vino, tipo_uva, porcentaje)

CP: {nom_vino, tipo_uva}

CAj: {nom_vino} --> Vino

El campo añada hace referencia al año que se vendimió la uva de ese vino, la bodega al almacén donde se elaboró, tipo_vino indica si se trata de tinto, blanco o rosado, tipo_uva puede ser garnacha, tempranillo, cariñena, etc. y porcentaje a la proporción en que esa uva está en ese vino (un vino puede estar formado por más de un tipo de uva). El resto de campos se sobreentienden.

Normaliza a 3FN las relaciones anteriores.

Solución:

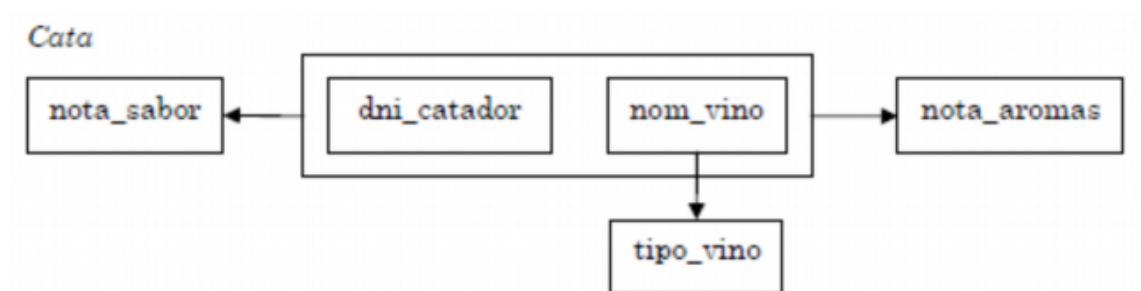
1FN

Como ninguna relación tiene atributos multivaluados o compuestos, ni derivados, todas las relaciones anteriores están en 1FN.

2FN

Hay que eliminar las DF parciales. Como las relaciones Catador y Vino no tienen claves principales compuestas, ya están en 2FN.

Para las relaciones Cata y Composición obtenemos el diagrama de DF para ver si hay dependencias parciales:



Como el atributo tipo_vino sólo depende de una parte de la clave, lo quitamos de la tabla Cata y lo agregamos en una nueva tabla llamada Tipo_vino.

Cata(dni_catador, nom_vino, nota_aromas, nota_sabor)

CP: {dni_catador, nom_vino}

CAj: {dni_catador} --> Catador

CAj: {nom_vino} --> Vino

Tipo_vino(nom_vino, tipo_vino)

CP: {nom_vino}

CAj: {nom_vino} --> Vino



Como el único atributo de la tabla depende de manera completa de la CP, la relación ya está en 2FN.

3FN

Hay que eliminar las DF transitivas. Para ello es necesario calcular el diagrama de DF de cada relación:

Catador(dni_catador, nombre, experiencia, dni_conyuge)

Como todos los atributos dependen directamente de la CP, la relación ya está en 3FN.

Vino(nom_vino, añada, precio, bodega)

Todos los atributos dependen directamente de la CP, luego la relación ya está en 3FN.

Cata(dni_catador, nom_vino, nota_aromas, nota_sabor)

Al depender todos los atributos de la CP la relación ya está en 3FN.

Tipo_vino(nom_vino, tipo_vino)

Como sólo hay un atributo, no puede haber dependencias transitivas, luego la relación ya está en 3FN.

Composición(nom_vino, tipo_uva, porcentaje)

Como la relación sólo tiene un atributo, no puede haber dependencias transitivas, luego la relación ya está en 3FN.

Así pues el conjunto final de relaciones en 3FN es el siguiente:

Catador(dni_catador, nombre, experiencia, dni_conyuge)

CP: {dni_catador}

Vino(nom_vino, añada, precio, bodega)

CP: {nom_vino}

Cata(dni_catador, nom_vino, nota_aromas, nota_sabor)

CP: {dni_catador, nom_vino}

CAj: {dni_catador} --> Catador

CAj: {nom_vino} --> Vino

Tipo_vino(nom_vino, tipo_vino)

CP: {nom_vino}

CAj: {nom_vino} --> Vino

Composición(nom_vino, tipo_uva, porcentaje)

CP: {nom_vino, tipo_uva}

CAj: {nom_vino} --> Vino