UD.3: MODELO LÓGICO.Parte 2 Normalización

Prácticas no evaluables
Boletín C (solucionado)

Bases de Datos (BD)
CFGS DAM/DAW

Abelardo Martínez y Pau Miñana.
Basado y modificado de Sergio Badal y Raquel Torres.
Curso 2023-2024

Aspectos a tener en cuenta

Estas actividades son opcionales y no evaluables pero es recomendable hacerlas para un mejor aprendizaje de la asignatura.

> 🚫 **Si buscas las soluciones por Internet o preguntas al oráculo de ChatGPT, te estarás engañando a ti mismo**. Ten en cuenta que ChatGPT no es infalible ni todopoderoso. Es una gran herramienta para agilizar el trabajo una vez se domina una materia, pero usarlo como atajo en el momento de adquirir habilidades y conocimientos básicos perjudica gravemente tu aprendizaje.
>
> Si lo utilizas para obtener soluciones o asesoramiento respecto a las tuyas, revisa cuidadosamente las soluciones propuestas igualmente. Intenta resolver las actividades utilizando los recursos que hemos visto y la documentación extendida que encontrarás en el "Aula Virtual".

ÍNDICE

1. Enunciado 1. Localidades

Normaliza a 3FN el siguiente modelo que relaciona las Provincias con sus Comarcas y sus Localidades.

Modelo Lógico:

```
Provincia (cod_prov, nom_prov)
CP: cod_prov

Comarca (cod_com, nom_com, cod_prov)
CP:  cod_com
CAj: cod_prov → Provincia {cod_prov}
VNN: cod_prov

Localidad (cod_loc, nom_loc, cod_com)
CP:  cod_loc
CAj: cod_com → Comarca {cod_com}
VNN: cod_com
```

Restricciones:

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

Solución

1FN

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

2FN

Hay que eliminar las dependencias funcionales (DF) parciales. Como ninguna de las relaciones anteriores tiene claves compuestas todas son completas, así que todas están en 2FN.

3FN

Hay que eliminar las DF transitivas. Para ello se puede realizar un diagrama de DF para cada relación y estudiar tabla por tabla, aunque en este caso resulta evidente que tanto los nombres como las claves foráneas incluidas no tienen dependencias transitivas. Ya está todo en 3NF.

Respecto a las restricciones, ya se cumplen por la propia definición de las tablas. Las claves foráneas indican precisamente relaciones 1:N donde cada localidad está en una comarca y cada comarca en una provincia, así que no hay nada que modificar al respecto.

En resumen, el enunciado proporciona un modelo relacional completamente definido y normalizado hasta 3FN. Rellenando con algunos datos de ejemplo se puede ver como quedaría:

PROVINCIA	
cod_prov	**nom_prov**
Castellón	12
Valencia	46
Alicante	3

COMARCA		
cod_com	**nom_com**	**cod_prov**
24	Vall d'albaida	46
27	L'alcoià	3
23	La Costera	46
8	alt Palancia	12

LOCALIDAD		
cod_loc	**nom_loc**	**cod_com**
46839	Guadasséquies	24
46830	Benigànim	24
3820	Cocentaina	27
12400	Segorbe	8

2. Enunciado 2. Cata de vinos

Normaliza a 3FN el siguiente modelo y define los dominios de los atributos. Ten en cuenta los siguientes detalles:

- El campo *añada* hace referencia al año que se vendimió la uva de ese vino.
- La *bodega* se refiere al almacén donde se elaboró.
- La *ubicación* es la dirección completa de la *bodega*.
- *n_aromas* y *n_sabor* son las notas en éstos aspectos, otorgadas en la cata.
- La *valoración* es la media de *n_aromas* y *n_sabor* entre todas las catas del vino.
- El campo *tipo_vino* indica si se trata de tinto, blanco o rosado.
- El campo *tipo_uva* puede ser garnacha, tempranillo, cariñena, etc.
- El *porcentaje* hace referencia a la proporción en que esa uva está en ese vino (un vino puede estar formado por más de un *tipo_uva*).
- El resto de campos se sobreentienden.

Modelo Lógico:

```
Catador (dni_catador, nombre, edad, fecha_nacimiento)
CP:  dni_catador
edad = Año_Actual – fecha_nacimiento

Vino (nombre, añada, precio, bodega, ubicación, valoración)
CP:  nombre

Cata (dni_catador, nom_vino, n_aromas, n_sabor, tipo_vino)
CP:  {dni_catador, nom_vino}
CAj: dni_catador → Catador {dni_catador}
CAj: nom_vino → Vino {nombre}

Composición (nom_vino, tipo_uva, porcentaje)
CP:  {nom_vino, tipo_uva}
CAj: nom_vino → Vino {nombre}
```

Solución

1FN

Comprobamos los atributos multivaluados, compuestos o derivados:

- Quitamos el atributo edad ya que se puede saber con la fecha de nacimiento (se podría añadir una restricción para forzar que el catador sea mayor de edad).
- La valoración se puede considerar calculada, aunque no responde realmente a una fórmula sino a una consulta sobre los datos en Cata (esto no aprenderemos a hacerlo hasta más adelante). Lo eliminamos a menos que se considere un dato lo suficientemente relevante y largo de re-calcular cada vez como para desnormalizar y mantenerlo.

```
Catador (dni_catador, nombre, fecha_nacimiento)
CP:  dni_catador
Restricción adicional:
fecha_actual-fecha_nacimiento> 18 años.

Vino (nombre, añada, precio, bodega, ubicación)
CP:  nombre
```

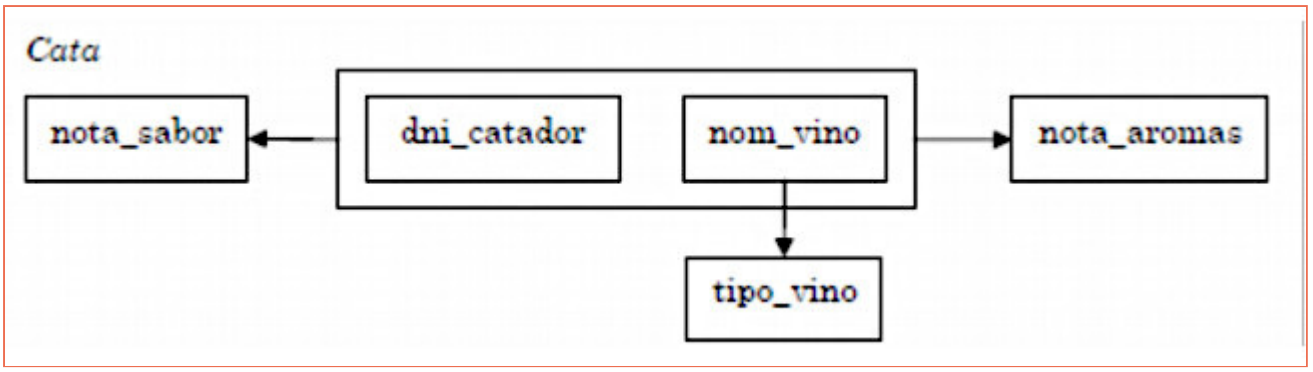
Ahora todas las relaciones están en 1FN.

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:

En Composición es inmediato ver que el porcentaje depende de la clave completa.



Como el atributo "tipo_vino" solo depende de una parte de la clave, lo quitamos de la tabla Cata y lo agregamos a la tabla Vino, pues la clave (el nombre del vino) coincide y no parece útil crear una nueva tabla.

```
Cata (dni_catador, nom_vino, n_aromas, n_sabor)
CP:  {dni_catador, nom_vino}
CAj: dni_catador → Catador {dni_catador}
CAj: nom_vino → Vino {nombre}

Vino (nombre, tipo_vino, añada, precio, bodega, ubicación)
CP: nombre
```

Ahora todas las relaciones están en 2FN.

3FN

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

Las siguientes relaciones no tienen dependencias transitivas:

```
Catador (dni_catador, nombre, fecha_nacimiento)
Cata (dni_catador, nom_vino, n_aromas, n_sabor)
Composición (nom_vino, tipo_uva, porcentaje)
```

En cambio en Vino la ubicación depende transitivamente de la bodega, así que la quitamos y separamos esta información en una nueva tabla Bodega:

```
Bodega (nombre_bod, ubicación)
CP:  nombre_bod

Vino (nombre, tipo_vino, añada, precio, bodega)
CP:  nombre
CAj: bodega → Bodega {nombre_bod}
```

Detectamos ciertas restricciones del dominio en los detalles del enunciado así que las añadimos al modelo, que está ahora normalizado en 3FN.

```
Catador (dni_catador, nombre, fecha_nacimiento)
CP:  dni_catador
Restricción adicional:
fecha_actual-fecha_nacimiento> 18 años.

Bodega (nombre_bod, ubicación)
CP:  nombre_bod

Vino (nombre, tipo_vino, añada, precio, bodega)
CP:  nombre
CAj: bodega → Bodega {nombre_bod}
Restricción adicional:
Valores para tipo_vino son: "tinto", "blanco" o "rosado".

Cata (dni_catador, nom_vino, n_aromas, n_sabor)
CP:  {dni_catador, nom_vino}
CAj: dni_catador → Catador {dni_catador}
CAj: nom_vino → Vino {nombre}
Restricción adicional:
n_aromas y n_sabor: valor de 0.0 a 10.0

Composición (nom_vino, tipo_uva, porcentaje)
CP:  {nom_vino, tipo_uva}
CAj: nom_vino → Vino {nombre}
Restricción adicional:
porcentaje: valor de 0.00 a 1.00
```

Para el tipo_uva no nos han proporcionado una lista de valores completa para poder restringir el dominio. Siempre se puede consultar con el cliente, incluso si puede que interese una tabla para los tipos de uva.

Si hay una definición de dominios, las restricciones añadidas pueden ir en esa definición, omitida por simplificación en estos boletines.

3. Enunciado 3. Proveedores

Normaliza a 3FN el siguiente modelo en el que se representan los artículos exportados por una serie de proveedores:

Modelo Lógico:

Artículo (cod_art, nombre, categoría, color, peso, descripción_categoría)
CP: cod_art

Proveedor (cod_prov, cod_art, nombre, {mail}^{n}, cantidad, sede.calle, sede.número, sede.ciudad, sede.país, {representante, nombre_rep}^{n})
CP: {cod_prov, cod_art}
CAj: cod_art → Artículo {cod_art}

Solución

1FN

En Proveedor movemos los campos multivaluados a sus respectivas tablas y separamos los compuestos, aunque realmente solo el representante esta expresado como tal, la sede simplemente comparte un prefijo que ayuda a la comprensión del atributo.

En ningún caso parece correcta la interpretación de que un representante trabaje para varios proveedores o el mail sea compartido, así pues las nuevas tablas se implementan con una debilidad de existencia, no de identificación.

Existen varias soluciones; con la información disponible sólo sabemos que el mail es multivaluado y tendría su propia tabla, pero vamos a suponer, ya que parece más lógico, que la expresión del modelo no era adecuada y los e-mails corresponden a los representantes y no al proveedor en si mismo. Suponemos además que cada representante tiene un solo mail, aunque esto no es en absoluto lo que el modelo estaba expresando.

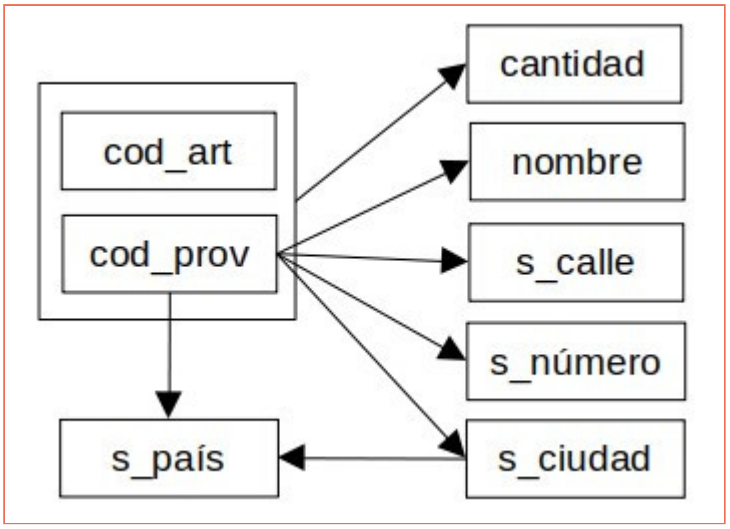
Ante un esquema así no quedaría más remedio que consultar las fuentes o al cliente para corroborar cual es la opción correcta. En caso que los mails correspondiesen a los representantes pero pudieran tener más de uno, se moverían a una tabla propia con los representantes como clave ajena.

```
Proveedor (cod_prov, cod_art, nombre, cantidad,
sede_calle, sede_número, sede_ciudad, sede_país)
CP:  {cod_prov, cod_art}
CAj: cod_art → Artículo {cod_art}

Representante (cod_rep, nombre_rep, mail, cod_prov*)
CP:  cod_rep
CAj: cod_prov → Proveedor {cod_prov}
VNN: cod_prov
```

2FN

Artículo y Representante tienen clave simple, ya están en 2FN. Comprobamos el diagrama de DF de Proveedor:



Con esto vemos que sólo la cantidad depende de la clave entera. Haremos lo siguiente:

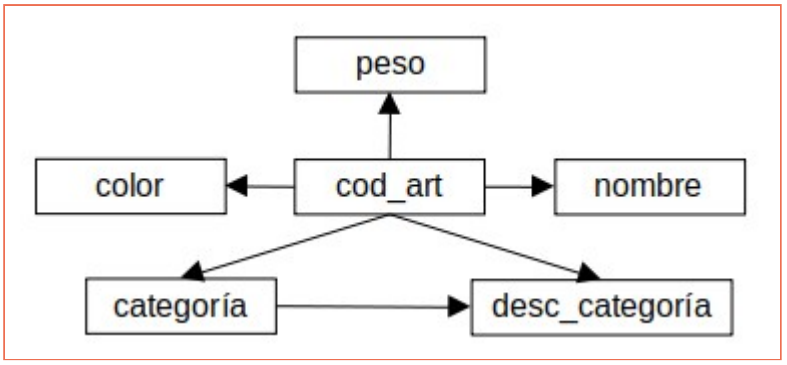
- En Proveedor dejamos la información que depende del mismo, con la clave simple cod_prov
- Creamos una relación Pedido con la clave compuesta {cod_prov, cod_art} y movemos allí la cantidad

```
Proveedor (cod_prov, nombre, sede_calle, sede_número,
sede_ciudad, sede_país)
CP:  cod_prov

Pedido (cod_prov, cod_art, cantidad)
CP:  {cod_prov, cod_art}
CAj: cod_prov → Proveedor {cod_prov}
CAj: cod_art → Artículo {cod_art}
```

3FN

- En Pedido no hay DF transitivas.
- En Proveedor, se puede ver en el diagrama anterior que sede_país depende transitivamente de la clave a través de sede_ciudad, así que creamos una tabla de ciudades y movemos allí el país.
- Si comprobamos el diagrama de DF de Artículo se ve fácilmente que laa descripción_categoría depende transitivamente a través de la categoría, con lo que se puede mover también a una tabla de categorías.



Con esto el resultado final en 3FN queda:

```
Categoría ( nombre_categoría, descripción_categoría)
CP:  nombre_categoría

Artículo (cod_art, nombre,  categoría, color, peso)
CP:  cod_art
CAj: categoría → Categoría {nombre_categoría}

Ciudad (nombre_ciudad, país)
CP:  nombre_ciudad

Proveedor (cod_prov, nombre, sede_calle, sede_número, sede_ciudad)
CP:  cod_prov
CAj: sede_ciudad → Ciudad {nombre_ciudad}

Representante (cod_rep, nombre_rep, mail, cod_prov*)
CP:  cod_rep
CAj: cod_prov → Proveedor {cod_prov}
VNN: cod_prov

Pedido (cod_prov, cod_art, cantidad)
CP:  {cod_prov, cod_art}
CAj: cod_prov → Proveedor {cod_prov}
CAj: cod_art → Artículo {cod_art}
```

4. Enunciado 4. Graduados

Normaliza a 3FN el siguiente modelo que representa los graduados en una serie de universidades.

Modelo Lógico:

```
Carrera (Id, Nombre, Créditos, Facultad, Nombre_facultad)
CP:  Id

Graduado (DNI, Nombre, Apellido, {Id_car}^n, Créditos)
CP:  DNI
CAj: Id_car → Carrera {Id}

Universidad (Id, Nombre, localidad, {Facultad}^n, {Id_car}^n)
CP:  Id
CAj: Id_car → Carrera {Id}
```

Solución

1FN

Aunque a primera vista no parece haber campos compuestos ni derivados, se puede observar un campo Créditos tanto en la Carrera como en el Graduado. Si conocemos los Créditos de cada Carrera y las carreras que tiene cada Graduado ya podemos calcular los Créditos Totales de cada Graduado sin necesidad de almacenar este dato, así que podemos considerarlo derivado y eliminarlo.

Veamos como actuar ahora en cada caso con los atributos multivaluados:

- En Graduado, Id_car se corresponde a la lista de carreras que puede tener cada uno. Evidentemente cada carrera puede tener muchos graduados, así que en la transformación consideramos una debilidad de identificación y la clave debe ser {DNI, Id_car} no solo Id_car. Se necesita una tabla nueva y en ningún caso se puede mover el DNI del Graduado a la Carrera.

  ```
  Graduado (DNI, Nombre, Apellido)
  CP:  DNI

  Grad_Carrera (DNI_grad, Id_car)
  CP:  {DNI_grad, Id_car}
  CAj: DNI_grad → Graduado {DNI}
  CAj: Id_car → Carrera {Id}
  ```

Es relativamente fácil ver que esto representa una relación N:M con los graduados en cada carrera, que o bien se había trasladado erróneamente al modelo relacional o no se había diseñado correctamente en el esquema E-R.

- En Universidad se da un caso parecido tanto con Id_car como con Facultad. Pero en este caso implementamos ambas debilidades como de existencia, al considerar que cada Facultad es propia de una única Universidad y cada Carrera también (aunque las carreras/títulos se consideran equivalentes, los planes de estudios no son iguales en todas las Universidades y en el título siempre se indica la Universidad de origen).

  Por esta razón, aunque se crea una tabla Facultad con esa clave (que llamaremos Id, por coherencia con las otras tablas), una tabla con la clave Id_car ya existe (Carrera), de modo que en ese caso solo se necesita poner allí como clave ajena la Id de la Universidad con restricción de valor no nulo.

```
Universidad (Id, Nombre, localidad)
CP:  Id

Facultad (Id, Id_uni*)
CP:  Id
CAj: Id_uni → Universidad {Id}
VNN: Id_uni

Carrera (Id, Nombre, Créditos, Facultad, Nombre_fac, Id_uni*)
CP:  Id
CAj: Id_uni → Universidad {Id}
VNN: Id:Uni
```

Con esto tenemos todas las tablas en 1NF.

2FN

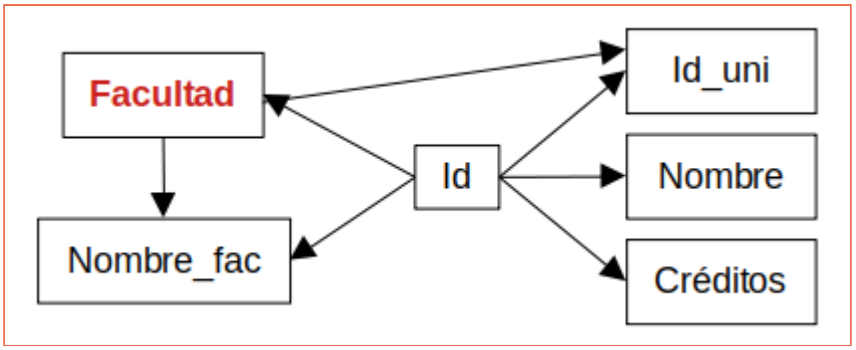
La Única tabla con clave compuesta es Grad_Carrera y no tiene otros campos, así que las tablas ya están en 2FN.

3FN

Analicemos las DF transitivas:

- Universidad, Facultad, Graduado y Grad_Carrera es evidente que no tienen.

- Veamos el diagrama de DF de Carrera:



Tanto el Nombre de la facultad como la Id_uni dependen transitivamente a través de la Facultad (recuerda que se ha decidido que cad Facultad pertenece a una única Universidad). Puesto que ya existe una tabla Facultad el Nombre_fac se mueve allí, pero la Id_uni ya se encuentra en esa tabla, es una información que ya tenemos y no otra relación distinta, con lo cual simplemente se elimina. Esto es lógico, pues si cada Carrera se estudia en una Facultad y éstas pertenecen a una Universidad ya se sabe en que Universidad se estudia cada Carrera, en esa tabla la referencia a la Universidad era redundante.

Este análisis debe servir además para ver que el campo Facultad es evidentemente una Clave Ajena apuntando a esa tabla, pero por el motivo que sea esto no estaba reflejado, de modo que lo incorporamos a nuestro modelo.

Se cambia además el nombre de la Clave Ajena Facultad a Id_fac por coherencia con el resto del modelo; puesto que todas las Carreras se estudian en alguna facultad, se añade además una restricción de valor no nulo.

```
Facultad (Id, Nombre_fac, Id_uni*)
CP:  Ud
CAj: Id_uni → Universidad {Id}
VNN: Id_uni

Carrera (Id, Nombre, Créditos, Id_fac*)
CP:  Id
CAj: Id_fac → Facultad {Id}
VNN: Id_fac
```

Con esto, el modelo normalizado a 3FN y claramente mejorado respecto una implementación inicial poco apropiada, queda del siguiente modo:

```
Universidad (Id, Nombre, localidad)
CP:  Id

Facultad (Id, Nombre_fac, Id_uni*)
CP:  Ud
CAj: Id_uni → Universidad {Id}
VNN: Id_uni

Carrera (Id, Nombre, Créditos, Id_fac*)
CP:  Id
CAj: Id_fac → Facultad {Id}
VNN: Id_fac

Graduado (DNI, Nombre, Apellido)
CP: DNI

Grad_Carrera (DNI_grad, Id_car)
CP:  {DNI_grad, Id_car}
CAj: DNI_grad → Graduado {DNI}
CAj: Id_car → Carrera {Id}
```

5. Bibliografía

- Iván López, M.ª Jesús Castellano. John Ospino. Bases de Datos. Ed. Garceta, 2a edición, 2017. ISBN: 978-8415452959
- Matilde Celma, Juan Carlos Casamayor y Laura Mota. Bases de datos relacionales. Ed. Prentice-Hall, 2003
- Cabrera Sánchez, Gregorio. Análisis y diseño detallado de aplicaciones informáticas de gestión. Ed. McGraw-Hill, 1st edition, 1999. ISBN: 8448122313