



TEMA 3. DISEÑO LÓGICO. NORMALIZACIÓN

EJERCICIOS SOKUCIONADOS

Bases de Datos
CFGs DAW

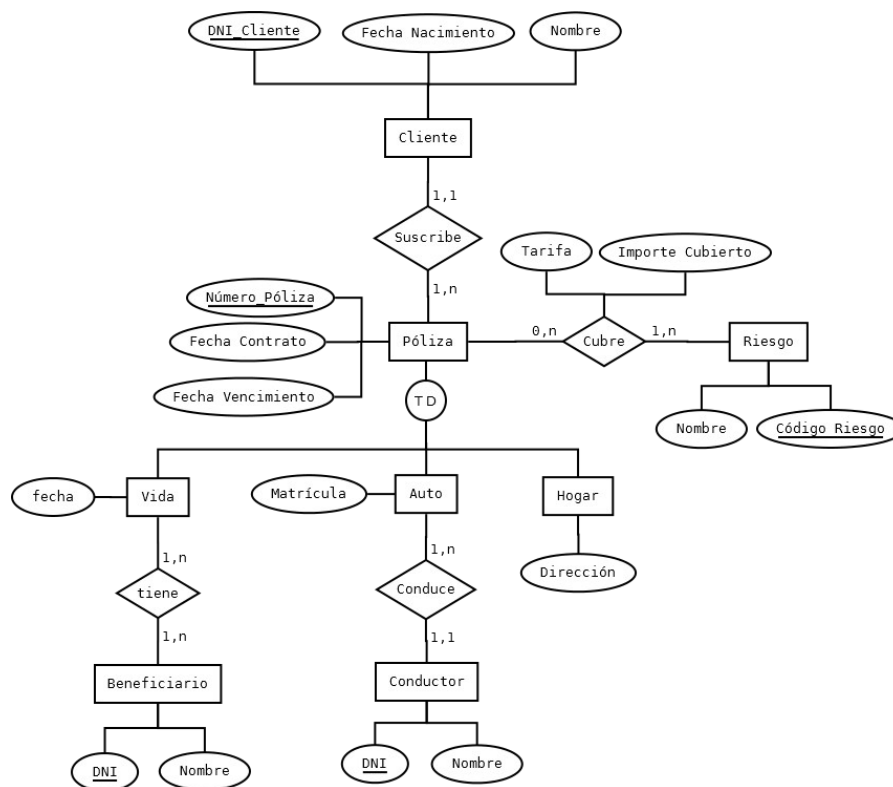
Recopilación de Ejercicios de:
Raquel Torres
Francisco Aldarias
Revisado por:
Pau Miñana Climent
2020/2021

UD03. DISEÑO LÓGICO. NORMALIZACIÓN

Transforma a Modelo Relacional los siguientes esquemas E/R. Las soluciones deben estar normalizadas a 3FN

EJERCICIO 1

El círculo con T,D es otra forma de especificar especializaciones/generalizaciones. Obviamente, T significa Total y D Disjunta.



Cliente (DNI_Cliente, Fecha_Nacimiento, Nombre_Cliente)

Conductor (DNI_Conductor, Nombre_Conductor)

Beneficiario (DNI_Beneficiario, Nombre_Beneficiario)

Riesgo (Cod_Riesgo, Nombre_Riesgo)

P_Vida (Num_Poliza, Fecha, F_Contrato, F_Vencimiento, DNI_Cliente)

P_Auto (Num_Poliza, Matrícula, F_Contrato, F_Vencimiento, DNI_Cliente)

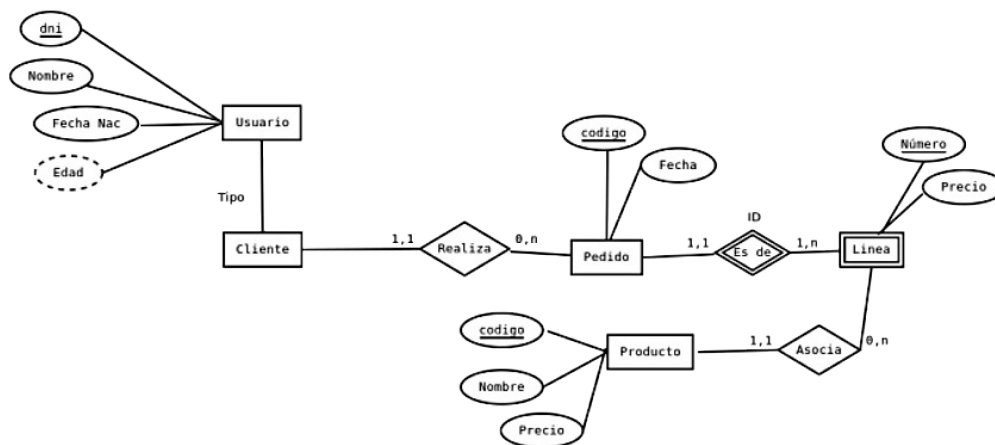
P_Hogar (Num_Poliza, Dirección, F_Contrato, F_Vencimiento, DNI_Cliente, DNI_Conductor)

Cubre (Num_Poliza, Cod_Riesgo, Tarifa, Importe_Cubierto)

Tiene (Num_Poliza, DNI_Beneficiario)

EJERCICIO 2

El rombo con línea doble implican una restricción de identificación, está marcado también cómo ID, así que no debería plantearos ninguna duda. Los atributos con la elipse en línea discontinúa indican campos derivados de otros; en este caso la edad se puede deducir de la Fecha de Nacimiento... Tener en cuenta al normalizar



Usuario (DNI, Nombre, Fecha_N, Fecha_Actual-Fecha_N)

Cliente (DNI_Cliente)

Pedido(Cod_Pedido, Fecha_P, DNI_Cliente)

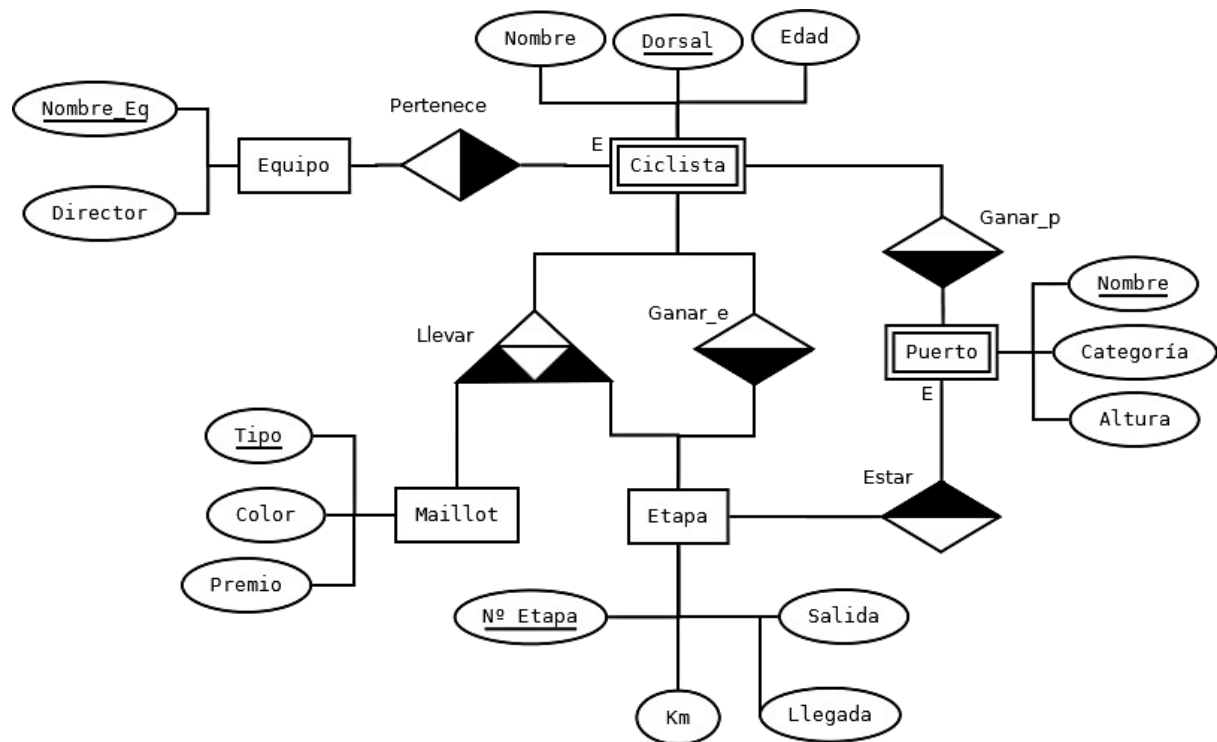
Producto (Cod_Producto, Nombre_Pr, Precio)

Línea (Número, Precio_Prod, Cod_Pedido, Cod_Producto)

Si os fijáis el esquema sigue una lógica algo extraña, parece que cada línea de pedido sólo sirve para pedir 1 unidad y el precio debería ser el del producto en sí mismo, lo que sería redundante y deberíamos eliminar al Normalizar, aunque puesto que es probable que la línea se use para imprimir las facturas tiene su lógica dejarlo. Una aproximación más lógica a este caso sería incluir una cantidad en la línea y precio como campo calculado (de nuevo, no lo quito por lógica a pesar de la normalización)

Línea (Número, Cod_Pedido, Cod_Producto, Cantidad, Cantidad*Precio_Prod)

EJERCICIO 3



Equipo (Nombre_Equipo, Director)

Ciclista (Dorsal, Nombre_C, Edad, Nombre_Equipo)

(Supondré edad campo normal pues no indican Fecha_NAC en ningún sitio)

Etapa (N_Etapa, Km, Salida, Llegada, Dorsal_Ganador)

Puerto (Nombre_P, Altura, Categoría, N_Etapa, Dorsal)

Maillot (Tipo, Premio, Color)

Llevar (Dorsal, N_Etapa, Tipo)

Normaliza a 3FN las siguientes relaciones

EJERCICIO 4

Artículo(cod_art, descrip, color, peso)

CP: {cod_art}

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

CP: {cod_prov, cod_art}

CAj: {cod_art} --> Artículo

Considera las siguientes dependencias funcionales (DF):

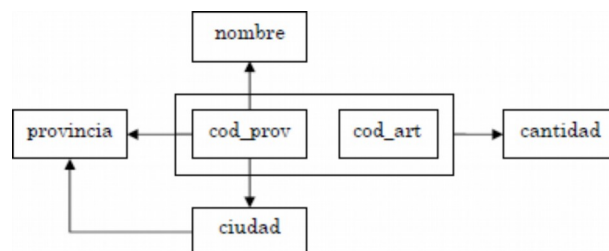
- cod_prov → nombre, provincia, ciudad
- cod_art → descripción, color, peso
- ciudad → provincia
- cod_prov, cod_art → cantidad

1FN

Puesto que no hay atributos compuestos ni multivaluados ya se encuentra en 1FN

2FN

Artículo tiene clave simple, ya se encuentra en 2FN. Veamos las dependencias en Proveedor:



Podemos ver que en realidad contiene información de proveedores, que sólo depende del código, y de artículos, en lo que parece un pedido que sí depende de la clave, así pues cambiamos la tabla de proveedor a una con clave simple y creamos Pedidos:

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

Eliminamos las DF transitivas, artículo y Pedidos no tienen ninguna, así que se quedan cómo están, en cambio en Proveedor la provincia depende de la ciudad, así que nos lo llevamos a una nueva tabla. Así pues el conjunto final de relaciones en 3FN es el siguiente:

Artículo (cod_art, descrip, color, peso)

Pedidos (cod_prov, cod_art, cantidad)

Proveedor (cod_prov, nombre, ciudad)

Ciudad (ciudad, provincia)

EJERCICIO 5

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, 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

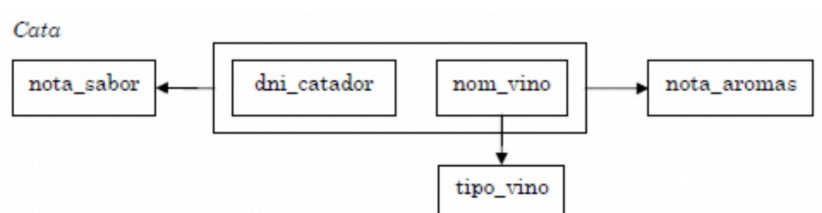
El campo *añada* hace referencia al año que se vendimió la uva de ese vino, la *bodega* al almacén donde se elaboró, *tipo_vino* indica si se trata de tinto, blanco o rosado, *tipo_uva* puede ser garnacha, tempranillo, cariñena, etc. y *porcentaje* a la proporción en que esa uva está en ese vino (un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

1FN

Puesto que no hay atributos compuestos ni multivaluados ya se encuentra en 1FN

2FN

Catador y vino no tienen claves compuestas, ya están en 2FN. Veamos las DF en cata y composición



El tipo_vino sólo depende del vino, así pues lo quitamos de cata y nos lo llevamos a Vino

Cata (dni_catador, nom_vino, nota_aromas, nota_sabor)

CP

Vino (nom_vino, añada, precio, bodega, tipo_vino)

(En caso de crear una tabla nueva debe observarse que tiene la misma clave que vino y un sólo atributo, luego no aporta nada tenerlo separado a nivel FN y deben juntarse o resulta redundante. A nivel de uso de la BD se puede considerar si una tabla Tipo_vino resultaría útil)

En Composición el único atributo depende de la clave entera así pues se encuentra en 2FN

3FN

Hay que eliminar las DF transitivas, pero un análisis de las dependencias nos permite ver que no existen y todos los atributos dependen directamente de la clave. Así pues el conjunto de relaciones siguiente ya se encuentra en 3FN

Catador (dni_catador, nombre, experiencia, dni_conyuge)

CP: {dni_catador}

Vino (nom_vino, añada, precio, bodega, tipo_vino)

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

Composición(nom_vino, tipo_uva, porcentaje)

CP: {nom_vino, tipo_uva}

CAj: {nom_vino} --> Vino

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.