



NORMALIZACION DE TABLAS

Y

DIAGRAMA ENTIDAD-RELACIÓN

Dra. I. Casanovas
Lic. M. Bertello
L. Palermo

INDICE

Normalización de tablas en el modelo relacional	3
Conceptos usados en el proceso de normalización	3
Primera Forma Normal (1FN)	5
Segunda Forma Normal (2FN)	8
Diagrama Entidad-Relación (DER)	14
Terminología	15
Relaciones y simbología para representarlas	15
Simbología para la construcción de un DER (Martin y Odell)	16
Construcción de un DER	18
Ejemplo guiado	19

NORMALIZACION DE TABLAS EN EL MODELO RELACIONAL

El proceso que valida que las tablas de una base de datos relacional estén bien estructuradas se llama normalización. La normalización es un proceso en el que un conjunto dado de relaciones es reemplazado por otro conjunto de relaciones, con una estructura más simple sin pérdida de información.

Las bases de datos relacionales se normalizan para:

- Evitar la redundancia de los datos.
- Evitar problemas de actualización de los datos en las tablas.
- Proteger la integridad de los datos.

La normalización se basa en el concepto de formas normales, denominadas: 1ª Forma Normal (1FN), 2ª forma normal (2FN), y 3ª forma normal (3FN).

Estas formas normales funcionan en forma anidada, es decir que para que una relación este en 3FN debe haber pasado por 2FN y esta por la 1FN.

Si bien existen más formas de normalización, estas primeras tres formas normales son suficientes para cubrir las necesidades de la mayoría de las bases de datos relacionales. El creador de estas 3 primeras formas normales (o reglas) fue Edgar F. Codd.

Conceptos usados en el proceso de normalización

- **Dependencia Funcional.-** es la relación que existe entre dos atributos. Ejemplo:

Dado un valor de X y un valor de Y, entonces Y es funcionalmente dependiente de X.

Un alumno es identificado por (depende de) su legajo

Legajo_alumno (X)	Nombre (Y)
001	Juan Pérez
002	Ana García

- **Clave primaria.-** Es el atributo que identifica unívocamente un registro de una tabla. No debe tener nunca un valor nulo. Generalmente se la indica subrayando el atributo clave. Este atributo hace que no tengamos filas (tuplas) repetidas. En el ejemplo, no debe haber en la tabla otro alumno con legajo 001.

<u>Legajo_alumno</u>	Nombre
001	Juan Perez
002	Ana Garcia
003	José Gonzalez
.....

Clave concatenada o compuesta

A veces es necesario más de un atributo para que funcione como clave primaria, en ese caso tenemos una clave primaria concatenada

Por ejemplo, para identificar unívocamente una factura no alcanza solo el número ya que una empresa puede emitir facturas tipo A y facturas tipo B y en cada una tendrá un ejemplar N° 1. En este caso se necesita el atributo Tipo de factura + N° de factura para tener una clave que identifique un único registro en la tabla Facturas

Clave Foránea

Cuando una clave primaria es usada como atributo en otra tabla se llama clave foránea. Es una referencia de ese atributo en otra tabla.

Por ejemplo: en una tabla de Clientes, la clave primaria puede ser Código de Cliente. En la tabla Pedidos, tendremos el N° de pedido como clave primaria, y comprenderá el Código de Cliente que realizó el pedido, los artículos y la cantidad solicitada, precio, etc. El atributo Código de Cliente, que es clave primaria en la tabla Clientes, es clave foránea en la tabla Pedidos.

La concordancia entre las claves foráneas y las claves primarias (el valor foráneo debe existir en la clave primaria) se llama integridad referencial y en el modelo relacional son la regla para establecer relaciones entre tablas.

Primera Forma Normal (1FN)

Los atributos deben tener valores atómicos o sea no pueden haber campos repetitivos

Cuando en el modelo de datos que nos proporcionan se nos indica que una entidad puede tener varios registros asociados, a este campo lo llamaremos campo repetitivo, es aquel que se encuentra indicado con un * al lado de su nombre y se indica entre paréntesis la cantidad de repeticiones que puede tener. En estas repeticiones se indicar el mínimo y el máximo.

Ejemplo: Supongamos que tenemos el siguiente modelo de datos para la tabla TRANSACCIONES de una empresa.

TRANSACCIONES	
<u>DNI</u>	
Nombre	
Dirección	
Correo Electrónico	
Teléfono Fijo *(0,3)	
VENTAS*(1,n)	
Nro Venta	
Fecha de Venta	
Nro de Producto	
Descripción Producto	
Cantidad Comprada	

donde el * indica la repetición. El primer número después del paréntesis indicará el número mínimo de repeticiones, pudiendo ser 0 o 1 y el número después de la coma indicará el número máximo de repeticiones admitidas. Pudiendo ser un número o la letra n que significara muchas repeticiones

Como ejemplo, los registros se verían así:

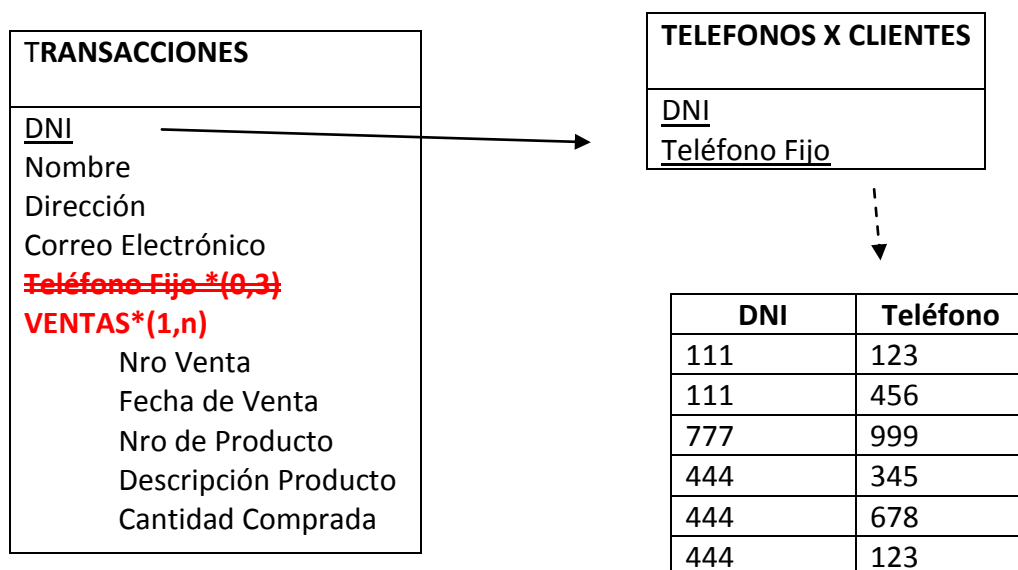
DNI	NOMBRE	DIRECCION	TE 1	TE 2	TE 3	Nº Venta	Fecha	Nº Prod.	Descrip	Cant	Nº Venta	Fecha	Nº Prod.	Descrip	Cant	Nº Venta	...
111	Prez	xxx	123	456		10	4-2-20	5	xxx	20	22	10-3-20	68	Xxxx	5		
777	Garcia	xxx	999			12	5-2-20	7	xxxx	10							
444	Lopez	xxxx	345	678	123	15	15-2-20	68	xxxx	30	35	15-3-20	54	xxx	25	40

En un esquema de archivos, podríamos fijar tres celdas para teléfonos ya que claramente se nos indica que ese es el máximo a registrar, pero en el caso de las ventas, pueden ser muchas, y el número de celdas del registro debe ser fijo, por lo que se debería estimar cuanto es "muchas". Como ejemplo, digamos 20...

Tenemos aquí el inconveniente de espacio desperdiciado (el cliente 777 solo realizó una transacción dejando el resto de celdas no utilizadas) o el caso inverso, un cliente que realice más de 20 transacciones nos generaría un problema en la tabla.

Aplicando la 1FN, se debe proceder a separar los campos repetitivos en una nueva tabla, y copiar en esta nueva tabla un campo que permite asociarlo a la entidad anterior para no perder la relación. En este ejemplo, tenemos dos casos de repetitividad: los Teléfonos (un Cliente puede tener registrados hasta 3 teléfonos) y las Ventas (un cliente puede tener muchas operaciones de venta)

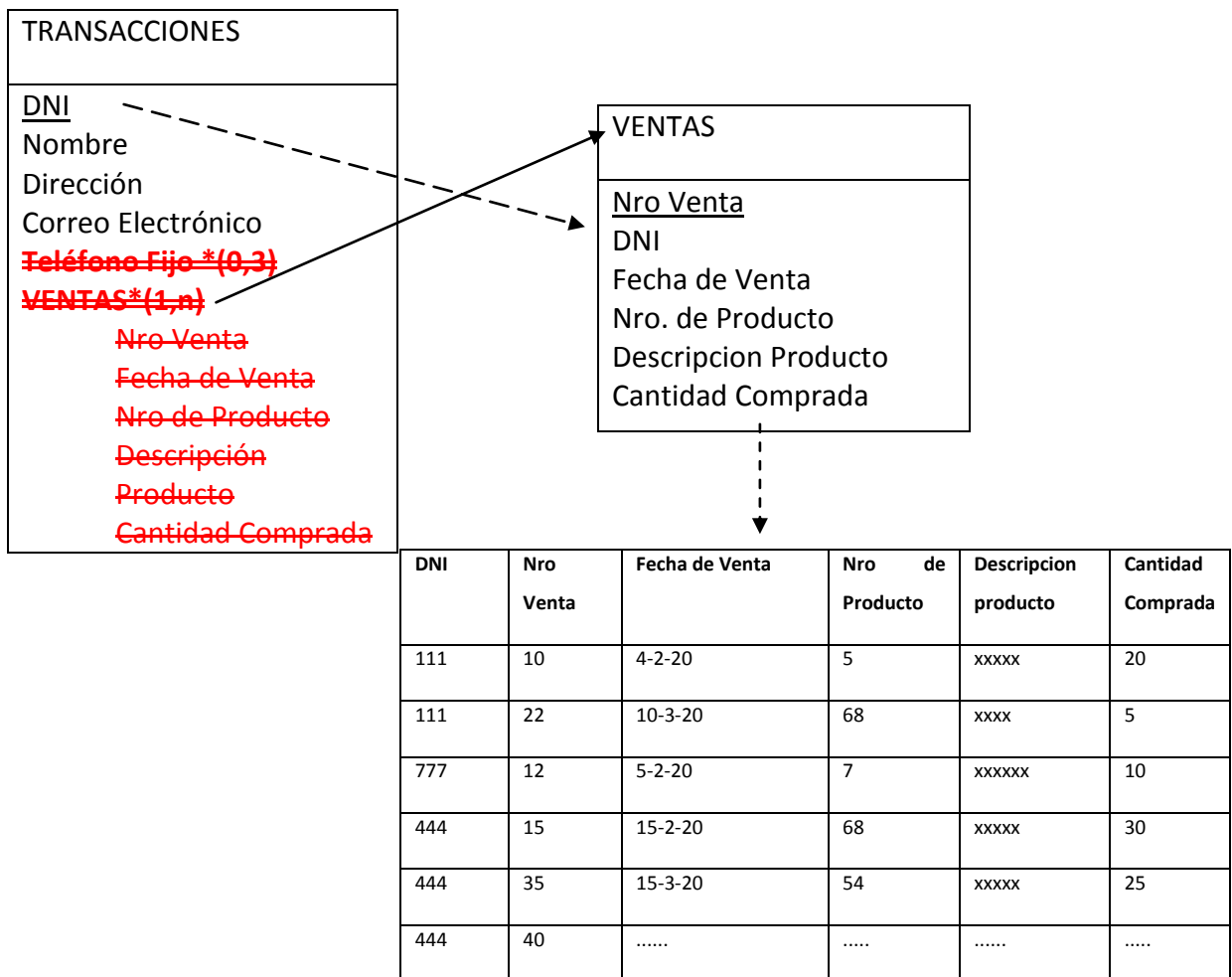
Desglosamos primero la entidad TELEFONOS x CLIENTES. El DNI será la referencia que vincule ambas tablas, pero no puede ser clave primaria simple ya que para un mismo DNI tendremos hasta 3 teléfonos, y el Nº de teléfono tampoco puede ser clave primaria simple ya que podría darse el caso que un mismo teléfono sea compartido por dos clientes, por lo tanto hay que crear una clave concatenada: DNI + Teléfono. La combinación de ambos atributos nos llevan siempre a un único registro de la tabla



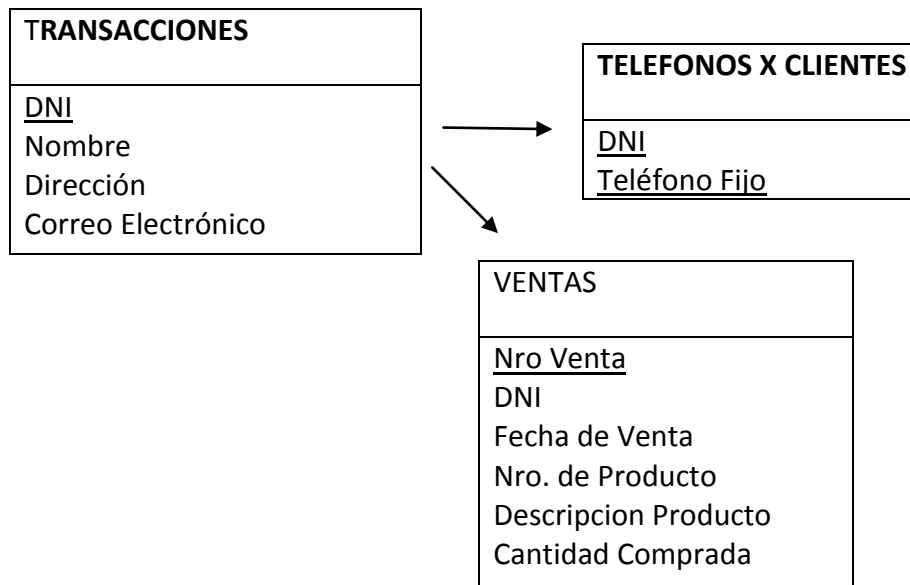
Resolvemos ahora la repetición de VENTAS.

Separamos la tabla Ventas, pero no podemos perder la información de a qué cliente pertenece esa venta, entonces guardamos la integridad referencial llevándonos el identificador único del cliente que realizó la transacción. Nro Venta es la clave primaria de la tabla Ventas ya que es un número secuencial correlativo, por lo tanto único, y DNI (en la tabla Ventas) es clave foránea que mantiene la conexión con los datos del cliente de la tabla Transacciones.

En nuestro ejemplo, nótese que VENTAS no es un campo si no que indica el nombre una estructura. Es decir, todos los campos que se encuentran dentro de esta estructura son los que se repiten en bloque.



Con lo cual, según nuestro ejemplo, para la primer forma normal, la tabla original quedaría desglosada en tres tablas:



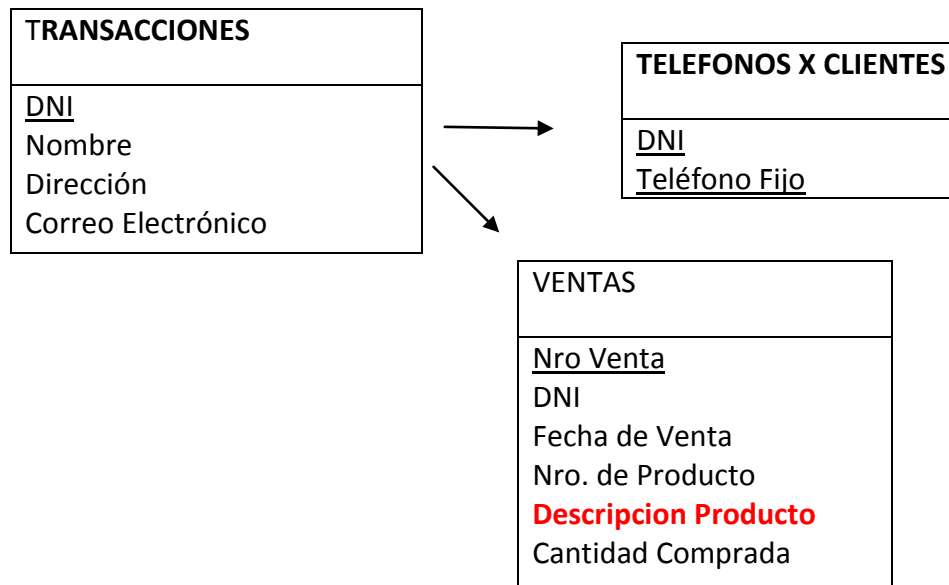
Segunda Forma Normal (2FN)

Una vez completada la primer forma normal del modelo de datos se pasa a la segunda forma normal.

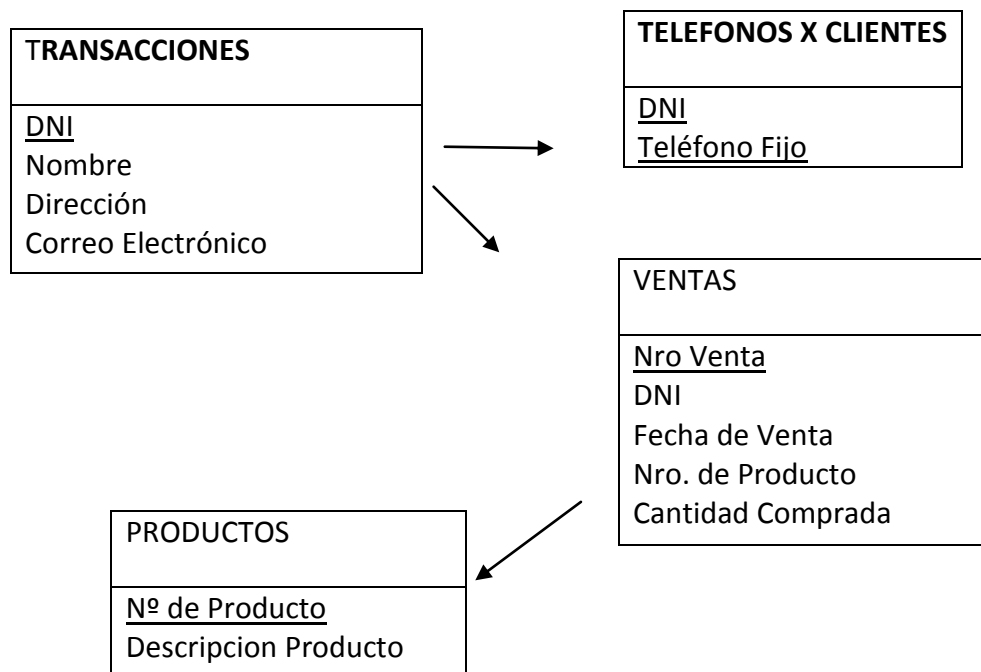
En la segunda forma normal se extraerán en una nueva tabla los atributos que no cumplan con la regla de dependencia funcional es decir que no estén directamente relacionados con la clave primaria (única o concatenada) de cada entidad que se elaboró a partir de la primera forma normal.

En el ejemplo anterior, ya en 1FN, identificamos donde no se cumple la regla de dependencia funcional:

- la Descripción del Producto no es un atributo que dependa del DNI y el N° de Venta. Depende del N° de Producto



Entonces debemos separarlo de la tabla VENTAS y armar la tabla PRODUCTOS, donde la clave primaria es el N° de Producto y Descripción depende funcionalmente de esa clave. De modo que toda la estructura en 2FN quedará:



Una vez que se complete la segunda forma normal, consideraremos que la estructura está normalizada. De una única estructura TRANSACCIONES, han quedado cuatro tablas normalizadas.

Se deberá corroborar que todas las tablas tengan distintas claves, no pudiendo tener dos o más tablas con iguales claves. Ejemplo: no puede haber dos o más tablas donde la clave sea solo DNI. Ya que si eso sucediera, o se definieron mal las claves o se trata de la misma tabla.

Con respecto a la **Tercera Forma Normal (3FN)**, en estos momentos está en desuso porque las restricciones que llevaron a crearla, ahora no existen. Veamos de qué se trata esta FN.

La 3FN indica que si un atributo es un campo "calculable" con los datos ya existentes en la tabla, debe omitirse. Por ejemplo, en una tabla de Facturas, deben necesariamente estar registrados la *Cantidad Vendida* del producto y el *Precio Unitario*, por lo que el *Importe Total*, resultante de multiplicar *Cantidad x Precio*, es un campo calculable que ocupa espacio de almacenamiento y debería omitirse. Cada vez que se necesitara conocer el Importe Total se procedía a realizar el cálculo pero no se guardaba el resultado. Esto era necesario cuando los equipos disponían de gran capacidad de procesamiento y poca capacidad de almacenamiento. Hoy en día sucede lo contrario, por lo que resulta más eficaz liberar al equipo de operaciones de cálculo y tener guardado el resultado, accesible directamente por operación de lectura.

¿Qué hacer si en el modelo de datos hay aclaraciones?

Una vez que se tenga la segunda forma normal completa, se deberá proceder a verificar que las aclaraciones (restricciones puestas por el usuario, generalmente debido a los registros existentes en la base a migrar) se cumplan. Si esto no sucede se deben modificar las tablas ya normalizadas de manera de hacer que las aclaraciones (restricciones) se cumplan.

EJEMPLO DE NORMALIZACIÓN ANIDADA

AEROPUERTOS DEL MUNDO

Nº País

Nº Ciudad

Cod. aeropuerto (EZE, MAD....)

Nombre del aeropuerto (Pistarini, Barajas, De Gaulle...)

Ubicación GPS

Página web

Cod. Tipo de aeropuerto

Desc Tipo de aeropuerto (internacional, nacional, militar....)

Terminales *(1,n)

Denominación (Term A, B, C...)

Transporte * (0,n)

Cod. Tipo Transporte

Descripción Tipo Transp (Bus, shuttle, tren...)

Horario de operación

Frecuencia

Lugar de parada

Instalaciones *(1,n)

Identificación fiscal de instalación

Cod. Tipo de Instalación

Descripción Tipo de Instalación (casa de cambio, cafetería....)

Nombre Comercial (McDonalds, Starbucks...)

Ubicación

Horario de atención

1= Argentina ; 2= España; 3
= Francia....

Ciudad 1 de Argentina=
Bs.As., Ciudad 2 de
Argentina = Córdoba...

Ciudad 1 de España =
Madrid; Ciudad 2 de
España = Barcelona....

Aclaraciones:

- Los códigos de Tipos de aeropuertos, de Tipo de Transporte, y de Tipo de Instalación son únicos para todos los aeropuertos

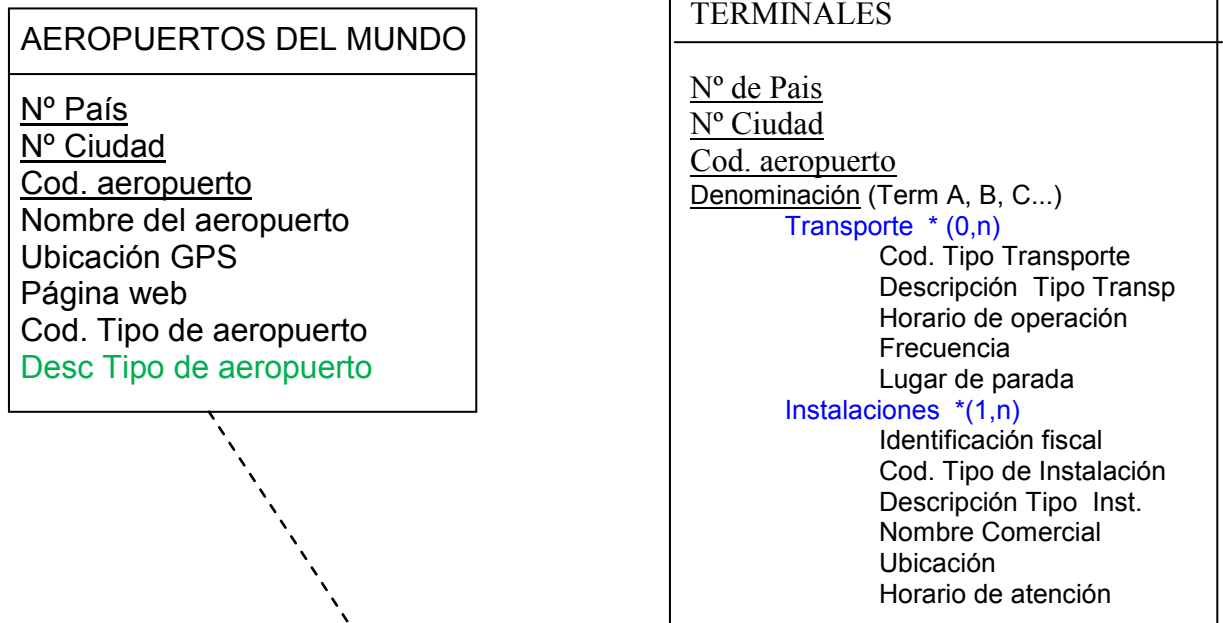
Razonamiento lógico:

Es necesario una clave concatenada ya que cada país tendrá una ciudad Nº 1, y cada ciudad puede tener más de un aeropuerto

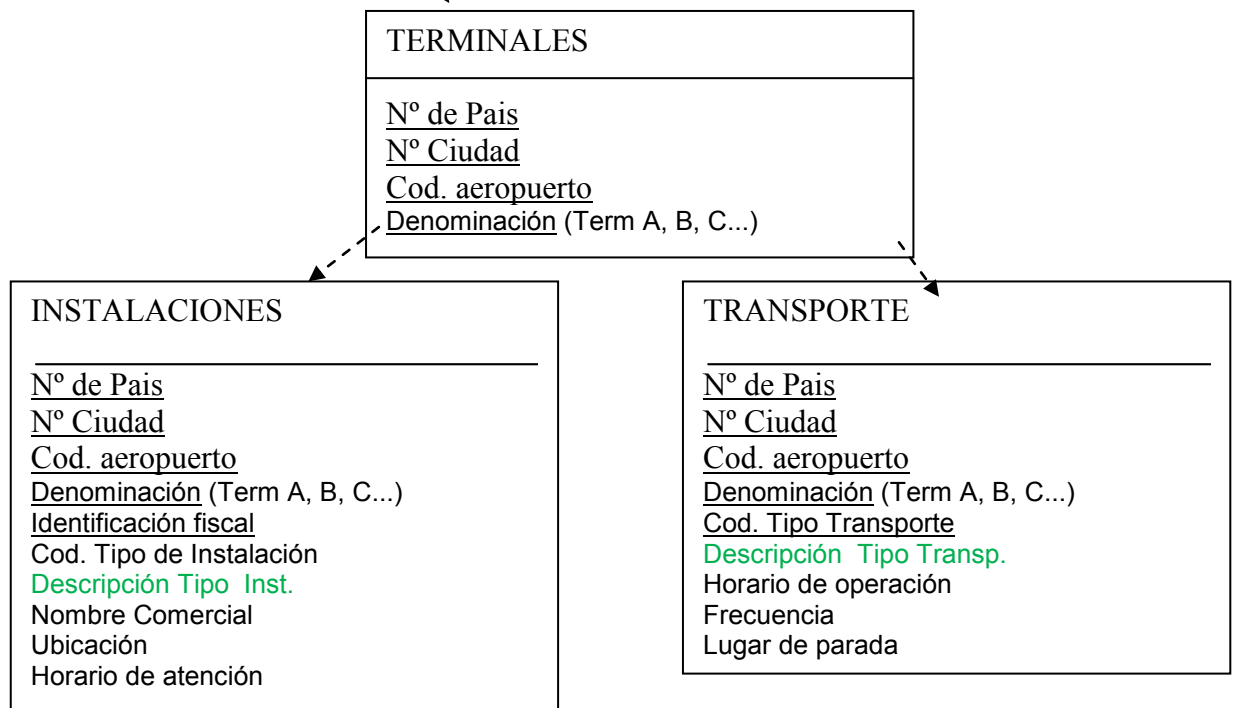
- > 1.1.EZE = Argentina + Bs.As.+ Pistarini
- > 1.1.AEP = Argentina + Bs.As. + Aeroparque
- > 1.2. COR = Argentina + Córdoba + Pajas Blancas
- > 2.2.BCN = España + Barcelona + El Prat
- > 3.1 CDG = Francia + Paris + Charles De Gaulle
- etc.etc.

Cada Aeropuerto puede tener como mínimo una terminal o varias.
 Cada Terminal tiene acceso por ninguno o varios medios de transporte
 Cada terminal tiene como mínimo una instalación o varias, que tienen un identificador único dentro de esa terminal

Por 1FN, resolvemos las repeticiones, primero y provisoriamente las que dependen de la clave principal o sea TERMINALES



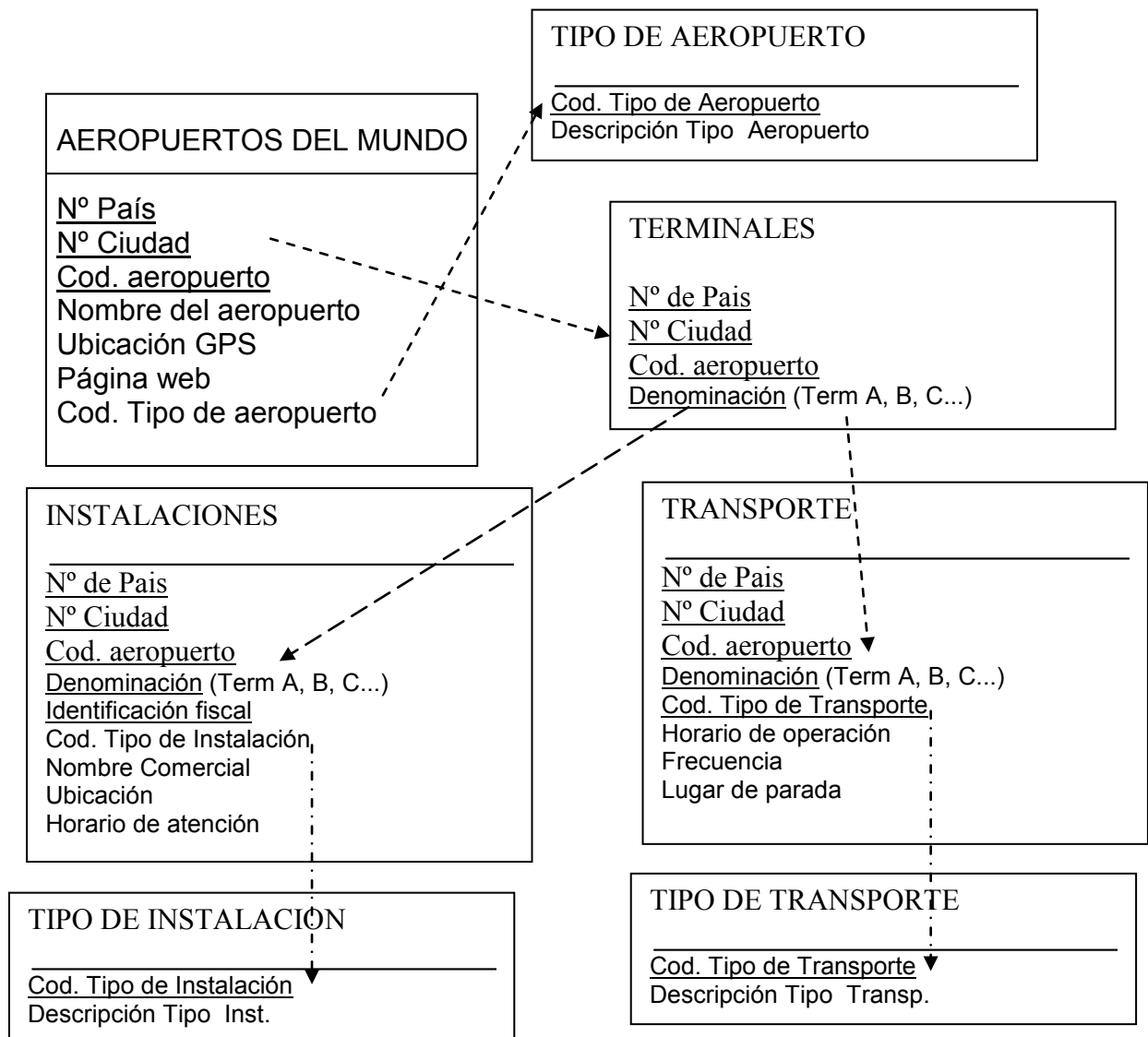
Pero dentro de la estructura terminales encontramos ahora dos estructuras que se repiten: TRANSPORTE e INSTALACIONES. Entonces la estructura TERMINALES debe ser a su vez normalizada y va a quedar:



Ya ha quedado la estructura original normalizada en 1FN. Debemos pasarla a la 2FN, eliminando las **dependencias no funcionales**.

- La Descripción del tipo de aeropuerto no depende del País-Ciudad sino solo del Código de Tipo de Aeropuerto
- De la misma forma, la Descripción del Tipo de Instalación depende de su código y la Descripción del Tipo de Transporte también

Entonces desglosamos estas dependencias en nuevas estructuras:



de modo que la versión final normalizada 1FN y 2FN de la estructura AEROPUERTOS DEL MUNDO quedaría como se ve en la siguiente página. Se ve cómo a través de la concatenación de claves no se pierde la conexión entre tablas. Faltaría agregar el Diagrama Entidad-Relación sobre estas tablas ya normalizadas.

Diagrama Entidad-Relación (DER)

Una vez finalizado el modelo esencial del análisis en el cual se estudió "qué" debe hacer el sistema, independientemente de la naturaleza de la tecnología que se usa para cubrir los requerimientos del usuario, la etapa siguiente es el diseño, en la cual se resuelve "cómo" se va a implementar lo planteado en la etapa de análisis.

Una de las tareas a realizar en esta etapa es la del diseño de los datos del sistema para lo cual se utiliza una herramienta de modelado de datos denominada diagrama entidad-relación (también conocido como D.E.R. o diagrama E-R). El D.E.R. representa los datos almacenados de un sistema como una red de almacenamientos conectados por relaciones, que deben estar previamente normalizados.

A Peter Chen de la Universidad Carnegie Mellon se le atribuye el desarrollo del modelo ER para el diseño de bases de datos. En los 70 publicó un documento influyente llamado "Modelo entidad-relación: hacia una visión unificada de los datos". Con posterioridad James Martin agregó mejoras al modelo de Chen.

Los Diagramas ER se usan para modelar y diseñar bases de datos relacionales, en términos de reglas de negocio y lógicas (en un modelo de datos lógicos). Se componen de entidades, relaciones y atributos. También representan la cardinalidad, que define las relaciones en términos numéricos. En la siguiente sección detallamos esta terminología

Se emplean un conjunto definido de símbolos para representar la interconexión de entidades, relaciones y sus atributos

Terminología del DER de diseño

- **Entidad:** representa una colección o conjunto de objetos (cosas) del mundo real cuyos miembros individuales (o instancias) tienen las siguientes características:
 - Cada una puede identificarse de manera única por algún medio.

- Cada uno puede describirse por uno o más datos (atributos).

Algo que se puede definir, como una persona, objeto, concepto u evento, que puede tener datos almacenados acerca del mismo. Por ejemplo, un Cliente, un Empleado, un Producto etc. Generalmente se grafican como un rectángulo.

- **Entidad asociativa:** entidad asociativa es aquella que relaciona entidades (o elementos) dentro de un conjunto de entidades. Por ejemplo, la entidad Clientes se relaciona con la entidad Productos a través de la entidad asociativa Pedidos ó Compras
- **Relación:** representa un conjunto de conexiones entre objetos (entidades). Por ejemplo, un Cliente compra un Producto
- **Atributo:** Una propiedad o característica de una entidad. Un Cliente tiene como atributos Nombre, Dirección, CUIL, etc. Un Producto tiene un Código, Descripción, Ubicación etc.

Relaciones y simbología para representarlas

Cardinalidad: Define los atributos numéricos, mínimos y máximos, de la relación entre dos entidades. Las tres relaciones cardinales principales son uno a uno, uno a muchos y muchos a muchos.

Una relación uno a uno es un vínculo entre la información de dos tablas, donde cada registro en cada tabla solo aparece una vez: una ocurrencia de en la entidad A no está en relación más que con una ocurrencia de la entidad B y, cada ocurrencia de B no está en

relación más que con una ocurrencia de A. Por ejemplo, puede haber una relación uno a uno entre un empleado y el vehículo que le ha dado la empresa para su actividad (un empleado tiene asignado un solo vehículo y cada vehículo lo maneja un único empleado).

En el caso de uno a muchos, una ocurrencia de la entidad A está en relación con una o muchas ocurrencias de la entidad B y, cada ocurrencia de B no está en relación más que con una ocurrencia de A. Encontramos la relación uno a muchos cuando una empresa tiene muchos empleados pero un empleado puede pertenecer solo a una empresa.

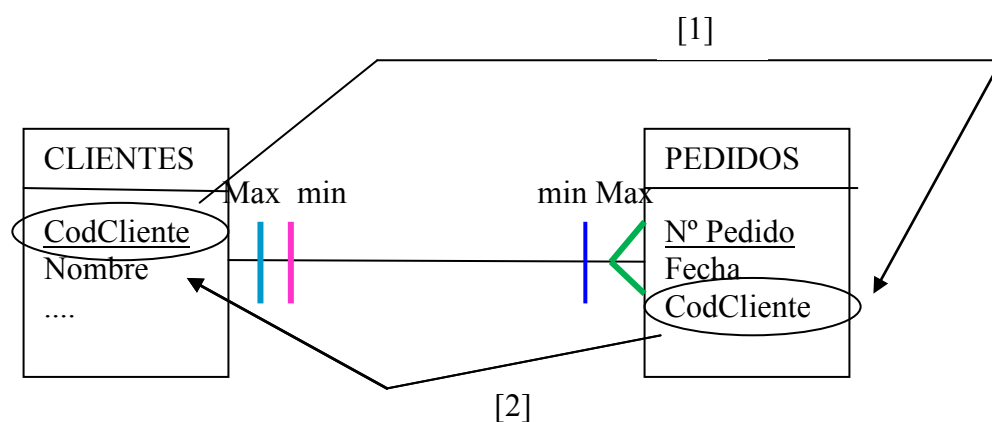
La relación muchos a muchos se da cuando una ocurrencia de la entidad A está en relación con una o muchas ocurrencias de B y, cada ocurrencia de la entidad B está en relación con una o muchas ocurrencias de A. Esta relación no está permitida en el modelo relacional y debe ser transformada mediante entidades asociativas. Un ejemplo de muchos a muchos sería un proveedor suministra muchos cursos y en un mismo producto pueden ser suministrado por varios proveedores.

Simbología para la construcción de un DER (Martin y Odell)

Para establecer la cardinalidad entre dos entidades, se procede de la siguiente forma:

Tomemos como ejemplo, establecer la cardinalidad de las relaciones entre la tabla CLIENTES y la tabla PEDIDOS

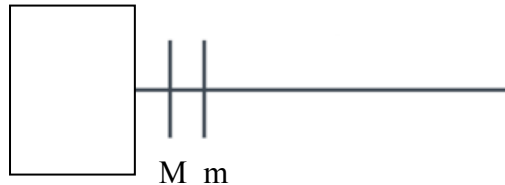
1. un Código de cliente (clave primaria) de la tabla CLIENTES puede aparecer como mínimo una vez en la tabla PEDIDOS (aceptando que para que esté registrado debe haber hecho algún pedido) y como máximo muchas veces (ya que puede haber realizado varios pedidos)
2. analizamos ahora la relación inversa: un Código de cliente (clave foránea en la tabla PEDIDOS) puede aparecer como mínimo una vez en la tabla CLIENTES (si se emitió el pedido es porque estaba registrado) y como máximo una vez (ya que una clave primaria solo puede aparecer una única vez en una tabla)



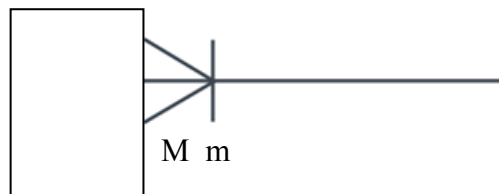
El máximo permitido de ocurrencia de una clave primaria (atributo) se grafica siempre más cercano al cuadro que representa la entidad y el mínimo, más alejado.

Recapitulando:

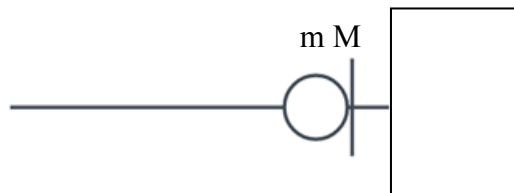
La cardinalidad máximo 1 y mínimo 1 se grafica:



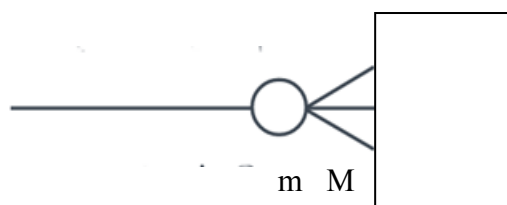
La cardinalidad máximo muchos y mínimo 1:



La cardinalidad mínima 0 y máximo 1: como ejemplo, una persona puede no realizar actividades deportivas en un equipo (participa individualmente) o puede pertenecer como máximo a un equipo.

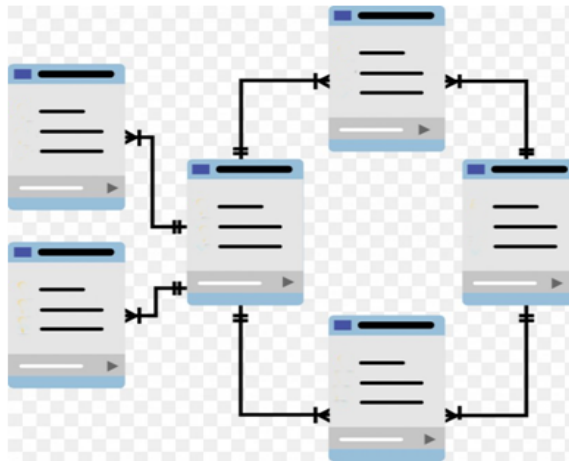


La cardinalidad mínima 0 y máxima muchos (un ejemplo de cardinalidad 0 lo tenemos en la relación entre la tabla PRODUCTOS y PEDIDOS: un Código de Producto, que existe en la tabla PRODUCTOS, puede no aparecer en la tabla PEDIDOS si ese producto, a pesar de estar en catálogo no tiene una venta, o puede ser vendido muchas veces:



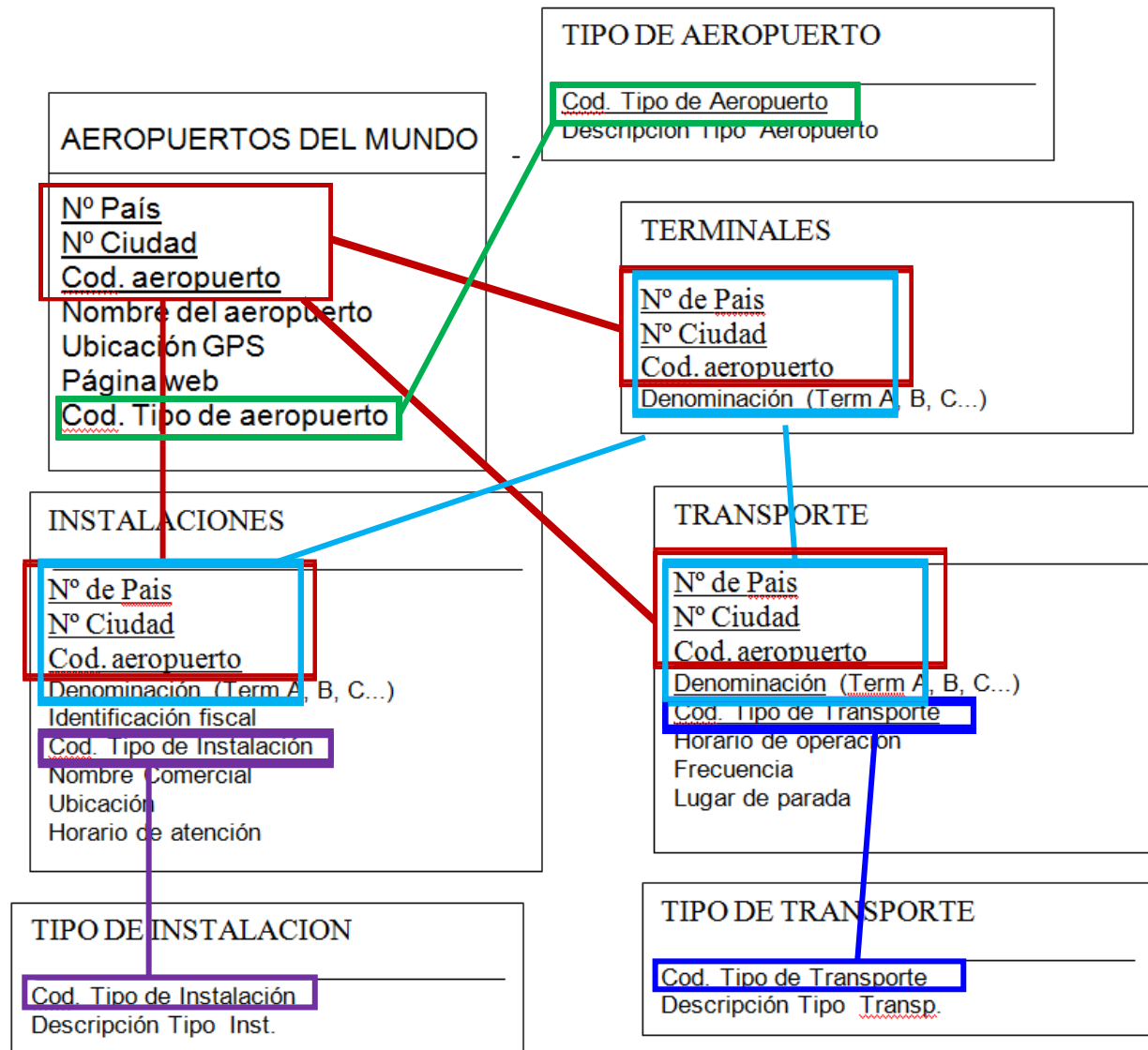
Construcción de un DER

Un diagrama entidad-relación, es un tipo de diagrama que ilustra cómo las "entidades" se relacionan entre sí dentro de un sistema. En el modelo relacional, estas entidades están representadas por tablas y deben relacionarse entre sí. No quiere decir que todas las tablas deben relacionarse con todas las tablas, sino que no debe quedar ninguna tabla desconectada de la red de relaciones.



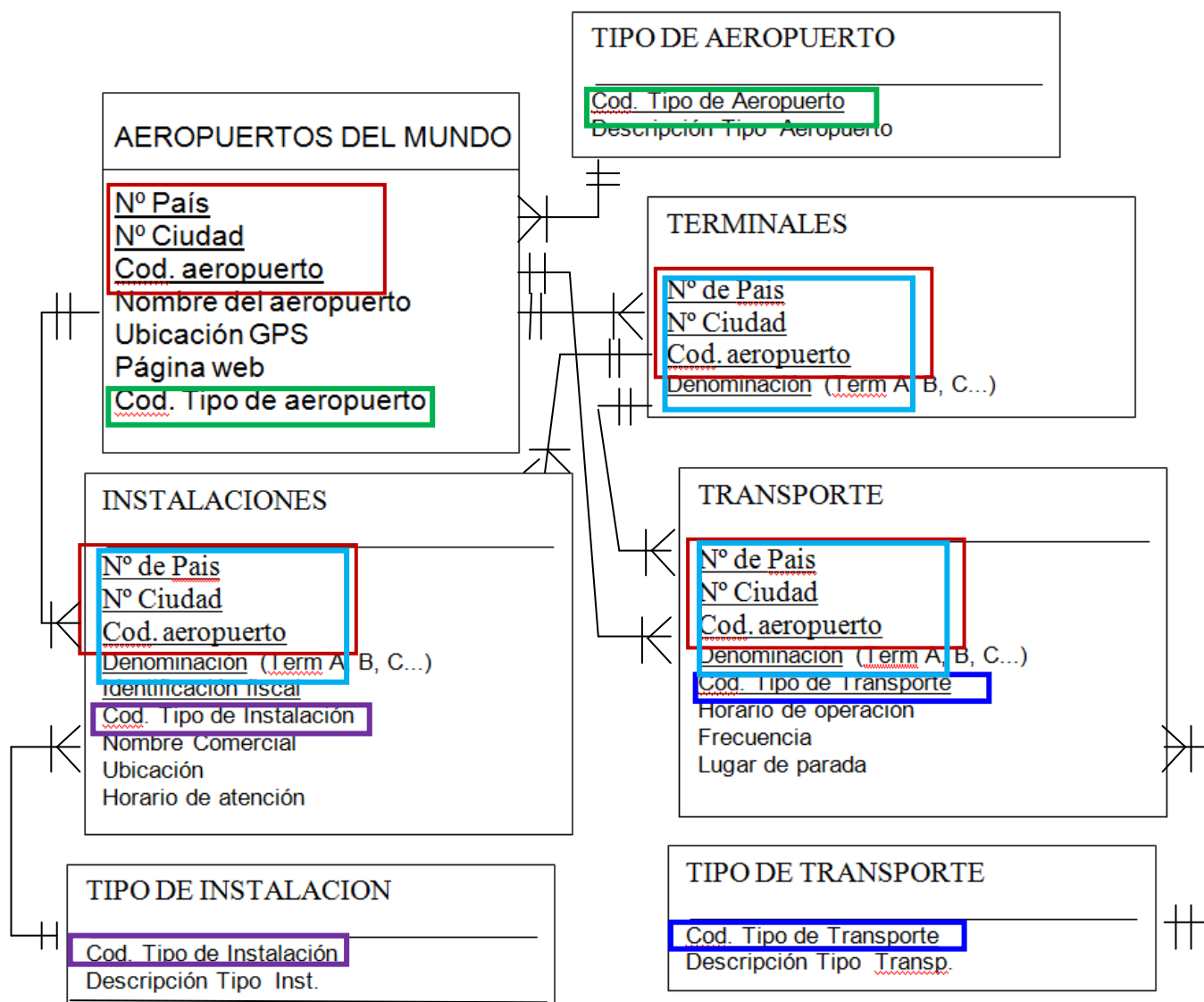
Para construir el diagrama, primero graficar las entidades (recordar que deben estar ya normalizadas) y sus atributos, subrayando las claves primarias. Luego, a partir de cada entidad y su clave primaria identificar donde aparecen como claves foráneas y diagramar las relaciones. Finalmente, analizar caso por caso, las cardinalidades.

EJEMPLO GUIADO



A partir de cada clave primaria voy conectando todas las tablas donde es clave foránea (parte de la clave concatenada o un atributo de la tabla)

1. En tabla AEROPUERTOS: Nº País+Nº Ciudad+ Cod.Aeropuerto son la clave primaria por la que iniciaremos la búsqueda de relaciones. Esa clave concatenada la encontramos como parte de la clave primaria de TERMINALES, en INSTALACIONES y en TRANSPORTE. Marcamos la relación, luego analizemos la cardinalidad.
2. En la tabla TERMINALES: : Nº País+Nº Ciudad+ Cod.Aeropuerto+ Denominación son la clave primaria por la que seguimos la búsqueda de relaciones. Esa clave concatenada la encontramos como parte de la clave primaria de TRANSPORTE e INSTALACIONES.
3. En la Tabla TIPO DE AEROPUERTO: Cod.Tipo Aeropuerto es clave primaria y es clave foránea en AEROPUERTOS
4. En la tabla TIPO DE INSTALACION: Cod. Tipo de Instalación es clave primaria y es clave foránea en INSTALACIONES
5. En la tabla TIPO DE TRANSPORTE: Cod. Tipo de Transporte es clave primaria y es clave foránea en TRANSPORTE.



Ahora analizamos las cardinalidades de las relaciones que encontramos en el paso anterior (se mantuvieron los colores para seguir mas fácil el ejemplo, no se dibujan en el DER) :

1. AEROPUERTOS-TERMINALES: un aeropuerto de una ciudad de un país puede tener 1 a muchas terminales. Una terminal de un aeropuerto de una ciudad de un país puede pertenecer solo a un aeropuerto
 2. AEROPUERTOS- INSTALACIONES: un aeropuerto de una ciudad de un país puede tener 1 a muchas instalaciones. Una instalación con una determinada identificación fiscal de un aeropuerto de una ciudad de un país puede pertenecer solo a un aeropuerto.
 3. AEROPUERTOS-TRANSPORTE: un aeropuerto de una ciudad de un país puede tener 1 a muchos transportes. Un determinado transporte de un aeropuerto de una ciudad de un país puede pertenecer solo a ese aeropuerto.
 4. TIPO AEROPUERTO-AEROPUERTO: un tipo de aeropuerto puede tener 1 a muchos aeropuertos (hay muchos aeropuertos civiles o militares por ej.) . Un código de tipo de aeropuerto puede existir una sola vez en Tipo de Aeropuertos (hay un único código para identificar cada tipo)
- Y así se siguen analizando todas las relaciones...Como la estructura ya estaba normalizada, no aparecen relaciones de muchos a muchos. Si aparecieran, debe revisarse la normalización.