

Normaliza hasta 3ª Forma normal la siguiente tabla que almacena suministros de artículos por parte de proveedores:

VIAJES (CONDUCTOR, MATRICULA, DNI, NOMBRE, KM, F_SALIDA, H_SALIDA, MARCA, C_MARCA, MODELO, COLOR, F_COMPRA, F_LLEGADA, H_LLEGADA, LUGAR_O, LUGAR_D, F_CARNET, F_NAC)

Donde los campos tienen el siguiente significado:

- CONDUCTOR: Código asignado al conductor
- MATRICULA Matrícula del vehículo
- DNI DNI del conductor
- NOMBRE Nombre del conductor
- KM Kilómetros realizados por un conductor y un vehículo en un viaje
- F_SALIDA Fecha de salida para un viaje
- H_SALIDA Hora de salida para un viaje
- MARCA Marca del vehículo. P.ej: Ford, SEAT, Mercedes,...
- C_MARCA Código de la marca del vehículo. P.ej: S (SEAT), F (Ford),...
- MODELO Modelo del vehículo. Ej: Marbella, Corsa, Toledo,...
- COLOR Color del vehículo
- F_COMPRA Fecha de compra del vehículo
- F_LLEGADA Fecha de llegada para cada viaje
- H_LLEGADA Hora de llegada para cada viaje
- LUGAR_O Lugar de origen del viaje.
- LUGAR_D Lugar de destino del viaje.
- F_CARNET Fecha en la que el conductor sacó el carnet de conducir.
- F_NAC Fecha de nacimiento del conductor.

Se tienen en cuenta las siguientes restricciones:

- Cada viaje tendrá un único conductor y se realizará en un único vehículo.
- Un conductor podrá utilizar diferentes vehículos, siempre en diferentes viajes.
- Un vehículo podrá ser utilizado para más de un viaje.
- La matrícula es única por vehículo.

En primer lugar, debemos comprobar en qué nivel de normalización nos encontramos (si es que la tabla ya está normalizada). Para ello debemos comprobar desde el nivel más bajo de normalización hacia el nivel superior.

Partimos de la siguiente relación:

VIAJES (CONDUCTOR, MATRICULA, DNI, NOMBRE, KM, F_SALIDA, H_SALIDA, MARCA, C_MARCA, MODELO, COLOR, F_COMPRA, F_LLEGADA, H_LLEGADA, LUGAR_O, LUGAR_D, F_CARNET, F_NAC)

Conductor	Matricula	DNI	Nombre	KMs	Fecha Salida	Hora Salida	Marca	Cod Marca	Modelo	Color	Fecha Compra	Fecha Llegada	Hora Llegada	Lugar Origen	Lugar Destino

1FN: Una base de datos se encuentra en primera forma normal si, y sólo si, en todas sus tablas, cada atributo es atómico.

No se encuentra en 1FN, porque encontramos varios atributos que no son atómicos.

Partimos de:

VIAJES (CONDUCTOR, MATRICULA, DNI, NOMBRE, KM, F_SALIDA, H_SALIDA, MARCA, C_MARCA, MODELO, COLOR, F_COMPRA, F_LLEGADA, H_LLEGADA, LUGAR_O, LUGAR_D, F_CARNET, F_NAC)

Analizamos que No se cumple la 1FN, ya que con un conductor y matrícula puede hacer viajes distintos (incluyendo KMs, fechas y horas de salida y llegada, lugar de origen y destino...)

Debemos transformar la relación llevando los atributos NO atómicos a una nueva tabla.

VIAJES (Conductor, Matrícula, DNI, Nombre, Fecha_Nacimiento, Fecha_Carnet, Cod_Marca, Marca, Modelo, Color, Fecha_Compra)

FECHA SALIDA (Conductor, Matrícula, Fecha_Salida, Hora_Salida, Fecha_Llegada, Hora_Llegada, Lugar_Origen, Lugar_Destino, KMs)

Por tanto, ahora se cumple la condición anterior y tanto las relaciones como la base de datos se encuentran en 1FN.

A continuación, comprobamos si la BBDD está ya en 2FN (para ello todas sus tablas tienen que estarlo)

2FN: Una base de datos está en 2FN si, y sólo sí, está en 1FN y además todos los atributos que no forman parte de la clave principal tienen dependencia funcional completa de ella.

Partimos de:

VIAJES (Conductor, Matrícula, Fecha_Salida, DNI, Nombre, Fecha_Nacimiento, Fecha_Carnet, KMs, Cod_Marca, Marca, Modelo, Color, Fecha_Compra)

FECHA SALIDA (Fecha_Salida, Hora_Salida, Fecha_Llegada, Hora_Llegada, Lugar_Origen, Lugar_Destino)

Analizamos las siguientes dependencias:

CONDUCTOR, MATRÍCULA → DNI, Nombre, Fecha_Nacimiento, Fecha_Carnet, KMs, Cod_Marca, Marca, Modelo, Color, Fecha_Compra

NO está en 2FN porque los atributos NO dependen de la clave completa.

- DNI, NOMBRE, FECHA_CARNET, y FECHA_NACIMIENTO no dependen de la matrícula, sino del conductor.
- COD_MARCA, MARCA, MODELO, COLOR no dependen del conductor, sino de la matrícula.
- TRAYECTO si está en 2FN, no necesita transformación.

De modo que se llegaría a las siguientes relaciones:

VIAJES (Conductor, Matrícula) (se puede quitar porque está repetido en la relación de TRAYECTOS)

TRAYECTOS (Conductor, Matrícula, Fecha_Salida, Hora_Salida, Fecha_Llegada, Hora_Llegada, Lugar_Origen, Lugar_Destino, KMs)

CONDUCTORES (Conductor, DNI, Nombre, Fecha_Nacimiento, Fecha_Carnet)

VEHÍCULO (Matrícula, Cod_Marca, Marca, Modelo, Color, Fecha_Compra)

Comprobamos que se cumple la condición, por lo que las relaciones ya están normalizadas a 2FN.

3FN: Una base de datos está en 3FN si, y sólo si, está en 2FN y además no existen dependencias transitivas. Todas las dependencias deben ser respecto de la clave principal.

Partimos de:

TRAYECTOS (Conductor, Matrícula, Fecha_Salida, Hora_Salida, Fecha_Llegada, Hora_Llegada, Lugar_Origen, Lugar_Destino, KMs)

CONDUCTORES (Conductor, DNI, Nombre, Fecha_Nacimiento, Fecha_Carnet)

VEHÍCULO (Matrícula, Cod_Marca, Marca, Modelo, Color, Fecha_Compra)

Analizando las dependencias funcionales anteriores detectamos transitividad en las siguientes dependencias:

DNI → Nombre, Fecha_Nacimiento

Cod_Marca → Marca

Por tanto, se debe crear una nueva relación para eliminar las dependencias transitivas, obteniendo el siguiente resultado en 3FN:

TRAYECTOS (Conductor, Matrícula, Fecha_Salida, Hora_Salida, Fecha_Llegada, Hora_Llegada, Lugar_Origen, Lugar_Destino, KMs)

CONDUCTORES (Conductor, DNI, Fecha_Carnet)

DNI (DNI, Nombre, Fecha_Nacimiento)

VEHÍCULO (Matrícula, Cod_Marca, Modelo, Color, Fecha_Compra)

MARCA (Cod_Marca, Marca)

Ahora la base de datos se encuentra normalizada en 3FN, ya que todas sus relaciones lo están.