

## Ejercicios de normalización

1. Normaliza a 3FN la siguiente relación, realizando los supuestos que consideres oportunos.

*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)*

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.

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

CONDUCTOR, MATRICULA → DNI, NOMBRE, MARCA, C\_MARCA, MODELO, COLOR, F\_CARNET, F\_NAC.

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

Debemos convertir los atributos multivaluados (llamado “no atómico”) en una nueva tabla.

VIAJES1(CONDUCTOR, MATRICULA, DNI, NOMBRE, MARCA, C\_MARCA, MODELO, COLOR, F\_COMPRA, F\_CARNET, F\_NAC)

VIAJES2(CONDUCTOR, MATRICULA, F\_SALIDA, H\_SALIDA, KM, F\_LLEGADA, H\_LLEGADA, LUGAR\_O, LUGAR\_D)

Ahora se puede afirmar que la base de datos está en 1FN.

A continuación, comprobamos si la base de datos está en 2FN (para ello todas sus tablas tienen que estarlo):

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

CONDUCTOR, MATRICULA → DNI, NOMBRE, MARCA, C\_MARCA, MODELO, COLOR, F\_CARNET, F\_NAC.

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

DNI, NOMBRE, F\_CARNET y F\_NAC no dependen de la matrícula sino del CONDUCTOR.

MARCA, C\_MARCA, MODELO y COLOR no dependen del conductor, dependen de la MATRÍCULA.

VIAJES2 está en 2FN, no necesita transformación.

Por tanto las relaciones se transforman de la siguiente manera, teniendo en cuenta que a la hora de establecer la clave primaria de una tabla debemos escoger un atributo o conjunto de atributos de los que dependan funcionalmente el resto de atributos.

VIAJES1(CONDUCTOR, MATRÍCULA)

CONDUCTOR(CONDUCTOR, DNI, NOMBRE, F\_CARNET, F\_NAC)

COCHE(MATRÍCULA, MARCA, C\_MARCA, MODELO, COLOR, F\_COMPRA)

VIAJES2(CONDUCTOR, MATRÍCULA, F\_SALIDA, H\_SALIDA, KM, F\_LLEGADA, H\_LLEGADA, LUGAR\_O, LUGAR\_D)

A continuación, comprobamos si la base de datos está en 3FN (para ello todas sus tablas tienen que estarlo):

**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.

Detectamos que en la relación coche, hay una relación transitiva:

- MATRÍCULA → C\_MARCA
- C\_MARCA → MARCA

Por lo tanto, la relación no está en 3FN y es necesario descomponerla de la siguiente forma.

COCHE(MATRÍCULA, C\_MARCA, MODELO, COLOR, F\_COMPRA)

MARCA(C\_MARCA, MARCA)

Finalmente, observamos que hay una redundancia entre VIAJES1 y VIAJES2, por lo que la primera relación está incluida en la segunda. La eliminamos y renombramos VIAJES2.

TRAYECTO (CONDUCTOR, MATRÍCULA, F\_SALIDA, H\_SALIDA, KM, F\_LLEGADA, H\_LLEGADA, LUGAR\_O, LUGAR\_D)

CONDUCTOR(CONDUCTOR, DNI, NOMBRE, F\_CARNET, F\_NAC)

COCHE(MATRÍCULA, C\_MARCA, MODELO, COLOR, F\_COMPRA)

MARCA(C\_MARCA, MARCA)