

```

viaje(cod,estado_convoy,min_demora){
    cod es una cadena no nula
    estado_convoy es una cadena
    min_demora es un numero
    PK(cod)
}
billete(cod,cod_viaje,metodo_pago,precio,asiento){
    cod es una cadena no nula
    cod_viaje es una cadena no nula
    metodo_pago es una cadena
    precio es un numero
    asiento es una cadena
    PK(cod,cod_viaje)
    FK(cod_viaje/viaje(cod))
}
pasajero(dni,sexo,nombre,edad,indicador_asistencia){
    dni es una cadena no nula
    cod_billete cadena no nula
    cod_viaje cadena no nula
    sexo es una cadena
    nombre es una cadena
    edad es un numero
    indicador_asistencia es una cadena
    PK(dni)
    FK(cod_billete,cod_viaje)/billete(cod,cod_viaje))

}
avala(dni_aval,dni_avalado){
    dni_aval es una cadena no nula
    dni_avalado es una cadena no nula
    PK(dni_aval)
    FK(dni_aval/pasajero(dni))
    FK(dni_avalado/pasajero(dni))
}
estacion(cod,ciudad,nombre,cod_post,num_vias,categoria){
    cod es una cadena no nula
    ciudad es una cadena
    cod_post es una cadena
    num_vias es un numero
    categoria es una cadena
    PK(cod)
}

```

```
realizan(cod_estacion,cod_viaje){  
    cod_estacion es una cadena no nula  
    cod_viaje es una cadena no nula  
    PK(cod_estacion,cod_viaje)  
    FK(cod_viaje/viaje(cod))  
    FK(cod_estacion/estacion(cod))  
}  
personal( dni,NSS,nombre,apellidos,antiguedad,salario_base,cod_estacion){  
    dni es cadena no nulo  
    NSS es cadena  
    nombre es cadena  
    apellidos es cadena  
    antiguedad es un numero  
    salario_base es un numero  
    cod_estacion es una cadena no nula  
    PK(dni)  
    FK(cod_estacion/estacion(cod))  
}  
  
conductor(DNI,licencia,experiencia){  
    DNI es cadena no nula  
    licencia es cadena  
    experiencia es un numero  
    PK(DNI)  
    FK(DNI/personal(dni))  
}  
vigilante(DNI,empresa_externa,num_placa){  
    DNI es cadena no nula  
    empresa_externa es cadena  
    num_placa es cadena  
    PK(DNI)  
    FK(DNI/personal(dni))  
}  
atencion_al_cliente(DNI,puesto,idioma_adicional){  
    DNI es cadena no nulo  
    puesto es cadena no nulo  
    idioma_adicional es cadena  
    PK(DNI)  
    FK(DNI/personal(dni))  
}  
ruta(cod,nombre,distancia_total,velocidad_maxima,cod_viaje){  
    cod es cadena no nulo  
    distancia_total es una cadena  
    velocidad_maxima es cadena
```

```

cod_viaje es cadena no nulo
PK(cod)
FK(cod_viaje/viaje(cod))
}

parada(cod_ruta,cod,via_llegada,tiempo_que_permanece){
    cod_ruta es cadena no nulo
    cod es cadena no nulo
    via_llegada es un numero
    tiempo_que_permanece es un numero
    PK(cod_ruta,cod)
    FK(cod_ruta/ruta(cod))
}

tren(matricula,modelo,fecha_fab){
    matricula es cadena no nulo
    modelo es cadena
    fecha_fab es date
    PK(matricula)
}

hacen(matricula,cod_parada,cod_ruta,dia_semana,hora_salida){
    matricula es cadena no nula
    cod_parada es cadena no nula
    cod_ruta es cadena no nulo
    dia_semana es cadena
    hora_salida es un numero
    PK(matricula,cod_parada,cod_ruta)
    FK(matricula/tren(matricula))
    FK(cod_parada,cod_ruta/parada(cod,cod_ruta))
}

revision(cod_revision,fecha,coste,tipo_reparacio,matricula){
    cod_revision es cadena no nulo
    fecha es date
    coste es numero
    matricula es cadena no nula
    tipo_reparacion es cadena
    PK(cod_revision)
    FK(matricula/tren(matricula))
}

```

```

vagones(matricula_tren,numero_bastidor,capacidad,servicios){
    matricula_tren es cadena no nulo
    numero_bastidor es un numero no nulo
    capacidad es un numero
    servicios es cadena
    PK(matricula_tren,numero_bastidor)
    FK(matricula_tren/tren(matricula))
}

```

## Formas normales

Para pasarlo a 1FN tenemos que ver que los valores de los atributos son atómicos al repasarlos la tabla estos serían los cambios:

```

estacion(cod,ciudad,nombre,cod_post,num_vias){
    cod es una cadena no nula
    ciudad es una cadena
    cod_post es una cadena
    num_vias es un numero
    PK(cod)
}
estacion_categorias(cod_estacion,categoria){
cod_estacion es una cadena no nula
categoria es una cadena
PK(cod_estacion)
FK(cod_estacion/estacion(cod))
}
atencion_al_cliente(DNI,puesto,){
    DNI es cadena no nulo
    puesto es cadena no nulo
    PK(DNI)
    FK(DNI/personal(dni))
}
atencion_al_cliente_idiomas(DNI,idiomas_adicional){
    DNI es cadena no nulo
    idiomas_adicional es cadena
    PK(DNI)
    FK(DNI/atencion_al_cliente(DNI))
}

```

```

vagones(matricula_tren,numero_bastidor,capacidad){
    matricula_tren es cadena no nulo
    numero_bastidor es un numero no nulo
    capacidad es un numero
    PK(matricula,numero_bastidor)
    FK(matricula/tren(matricula))
}
vagones_servicios(matricula_tren,numero_bastidor,servicio){
    matricula_tren es cadena no nulo
    numero_bastidor es un numero no nulo
    servicios es cadena
    PK(matricula,numero_bastidor)
    FK(matricula_tren/vagones(Matricula_tren))
    FK2(matricula,numero_bastidor/vagones(matricula,numero_bastidor)) }

```

Para pasarlo a 2FN previamente debe de estar en 1FN y además todos los atributos no clave dependen de la clave primaria en su totalidad estos serían los cambios:

Analizaremos solo las tablas con clave primaria compuesta que son :

- billete(cod, cod\_viaje, metodo\_pago, precio, asiento)
- realizan(cod\_estacion, cod\_viaje)
- parada(cod\_ruta, cod, via\_llegada, tiempo\_que\_permanece)
- hacen(matricula, cod\_parada, cod\_ruta, dia\_semana, hora\_salida)
- vagones(matricula\_tren, numero\_bastidor, capacidad)

En todas como podemos ver los atributos no clave dependen de la clave primaria completa, no de una parte y no existen dependencias parciales.

Para pasarlo a 3FN previamente debe estar en 2FN y además no existen dependencias transitivas entre atributos no clave (significa que los atributos no clave no dependan de otros atributos que no sean clave)

Tras analizarlo, estos son los cambios necesarios para conseguirlo :

```

estación (cod,nombre,cod_post, num_vias){
    cod es una cadena no nula
    nombre es una cadena
    cod_post es una cadena no nula
    num_vias es un número

    PK(cod)

```

```

        FK(cod_post / codigos_postales (cod_post))
    }
codigos_postales (cod_post, ciudad){
    cod_post es cadena no nulo
    ciudad es cadena,
}

PK(cod_post)
}

```

## RESUMEN :

1FN: Eliminamos las listas en servicios, categorias e idiomas.

2FN: Aseguramos que en las tablas con claves compuestas (como parada o hacen), los datos dependan de toda la clave.

3FN: Separamos la ciudad del codigo\_postal para evitar redundancia si varias estaciones están en la misma ciudad.

Asi quedarian todas las tablas :

- codigos\_postales (cod\_post, ciudad)
- estacion (cod, nombre, cod\_post, num\_vias)
- estacion\_categorias (cod\_estacion, categoria)
- realizan (cod\_estacion, cod\_viaje)
- personal (dni, NSS, nombre, apellidos, antiguedad, salario\_base, cod\_estacion)
- conductor (DNI, licencia, experiencia)
- vigilante (DNI, empresa\_externa, num\_placa)
- atencion\_al\_cliente (DNI, puesto)
- idiomas\_personal (DNI, idioma)
- viaje (cod, estado\_convoy, min\_demora)
- tren (matricula, modelo, fecha\_fab)

- ruta (cod, nombre, distancia\_total, velocidad\_maxima, cod\_viaje)
  - parada (cod\_ruta, cod, via\_llegada, tiempo\_que\_permanece)
  - hacen (matricula, cod\_parada, cod\_ruta, dia\_semana, hora\_salida)
  - vagones (matricula\_tren, numero\_bastidor, capacidad)
  - vagon\_servicios (matricula\_tren, numero\_bastidor, servicio)
  - revision (cod\_revision, fecha, coste, tipo\_reparacion, matricula\_tren)
- 
- pasajero (dni, sexo, nombre, edad, indicador\_asistencia)
  - avala (dni\_avala, dni\_avalado)
  - billete (cod, cod\_viaje, metodo\_pago, precio, asiento)