

## 2) DFs:

1. #poliza -> tipo\_seguro, f\_inicio, f\_fin, #cliente
2. #poliza -> tipo\_seguro, f\_inicio, f\_fin, cuil
3. #cliente -> cuil, nombre, dir, tel
4. cuil -> cliente, nombre, dir, tel
5. #siniestro -> f\_denuncia, estado\_sin

Clave candidata: {#poliza, #siniestro, #perito, #cuota, detalle\_reparacion}

POLIZAS (#poliza, #cliente, cuil, nombre, dir, tel, tipo\_seguro, f\_inicio, f\_fin, #siniestro, f\_denuncia, estado\_sin, #perito, #cuota, detalle\_reparacion)

DF1 y DF2 son equivalentes, ya que producen el mismo conjunto de atributos.

DF6 y DF7 son equivalentes entre sí.

POLIZAS no está en BCFN porque existe, al menos, {#cliente} de la DF3 que NO es superclave del esquema.

Iteración 1: particiono POLIZAS, siguiendo la DF3:

L1(#cliente, cuil, nombre, dir, tel)

L2(polizas - { cuil, nombre, dir, tel } =

L2 (#poliza, #cliente, tipo\_seguro, f\_inicio, f\_fin, #siniestro, f\_denuncia, estado\_sin, #perito, #cuota, detalle\_reparacion)

$L1 \cap L2 = \text{\#cliente}$ , que es clave en L1. Por lo tanto, no se perdió información.

Por **validación simple**, verifico que no se pierden dependencias funcionales porque:

L1 (vale): (DF) 3

L2: 1, 5

L1 está en BCNF ya que {#cliente} es superclave del esquema y sólo vale la DF3 en el esquema.

L2 no está en BCFN porque existe, al menos, {#poliza} de la DF1 que NO es superclave del esquema.

La DF2 y DF4 no se pierden, ya que es posible acceder a todos sus determinados -> #cliente existe en L2, y debido a la equivalencia entre DF3 y DF4, con #cliente se puede acceder a cuil, lo que demuestra que estas dependencias siguen vigentes -> no se pierden dependencias funcionales.

Iteración 2: particiono L2, siguiendo la DF1:

L1(#cliente, cuil, nombre, dir, tel)

L3(#poliza, tipo\_seguro, f\_inicio, f\_fin, #cliente)

L4(L2 - { tipo\_seguro, f\_inicio, f\_fin, #cliente } =

L4 (#poliza, #siniestro, f\_denuncia, estado\_sin, #perito, #cuota, detalle\_reparacion)

$L3 \cap L4 = \#poliza$ , que es clave en L3. Por lo tanto, no se perdió información.

Por **validación simple**, verifico que no se pierden dependencias funcionales porque:

L1 (vale): (DF) 3

L3: 1

L4: 5

L3 está en BCNF ya que {#poliza} es superclave del esquema y sólo vale la DF1 en el esquema.

L4 no está en BCFN porque existe, al menos, {#siniestro} de la DF5 que NO es superclave del esquema.

Iteración 3: particiono L4, siguiendo la DF5:

L1 (#cliente, cuil, nombre, dir, tel)

L3 (#poliza, tipo\_seguro, f\_inicio, f\_fin, #cliente)

L5 (#siniestro, f\_denuncia, estado\_sin)

L6(L4 - { f\_denuncia, estado\_sin } =

L6 (#poliza, #siniestro, #perito, #cuota, detalle\_reparacion)

$L5 \cap L6 = \text{\#siniestro}$ , que es clave en L5. Por lo tanto, no se perdió información.

Por **validación simple**, verifico que no se pierden dependencias funcionales porque:

L1 (vale): (DF) 3

L3: 1

L5: 5

L3 está en BCNF ya que {#siniestro} es superclave del esquema y sólo vale la DF5 en el esquema.

**L4 cumple con BCFN puesto que cualquier DF que se pueda encontrar ahí es trivial -> tiene los mismos atributos que la CC.**

## Normalización a BCNF.

Se terminó BCNF con las siguientes particiones:

L1 (#cliente, cuil, nombre, dir, tel)

L3 (#poliza, tipo\_seguro, f\_inicio, f\_fin, #cliente)

L5 (#siniestro, f\_denuncia, estado\_sin)

L6 (#poliza, #siniestro, #perito, #cuota, detalle\_reparacion)

**Clave primaria:** (#poliza, #siniestro, #perito, #cuota, detalle\_reparacion)

## Normalización a 4FN.

Se encontraron las siguientes DF multivaluadas en L6:

1. #poliza --> #cuota
2. #siniestro --> #perito
3. #poliza, #siniestro --> detalle\_reparacion NO-> ~~#poliza~~ --> ~~#siniestro~~

Por lo tanto, el esquema L8 no está en 4FN porque existe al menos una DM -> ej DM3 que no es trivial en el esquema.

Teniendo en cuenta DM1:

L7(#poliza, #cuota)

L8(#poliza, #siniestro, detalle\_reparacion, #perito)

L7 está en 4FN porque no valen DMs que no sean triviales en ella.

L8 no está en 4NF porque hay al menos una DM, ej DM2 que no es trivial en L8.

Considerando DM2:

L9(#siniestro, #perito)

L10(#poliza, #siniestro, detalle\_reparacion)

Tanto L9 como L10 están en 4FN ya que no existen DM que no sean triviales en ellas.

Esquemas resultantes en 4FN y que no son proyecciones de otros:

L1 (#cliente, cuil, nombre, dir, tel) -> porque no hay multivaluadas

L3 (#poliza, tipo\_seguro, f\_inicio, f\_fin, #cliente)

L5 (#siniestro, f\_denuncia, estado\_sin)

L7 (#poliza, #cuota)

L9 (#siniestro, #perito)

L10 (#poliza, #siniestro, detalle\_reparacion)