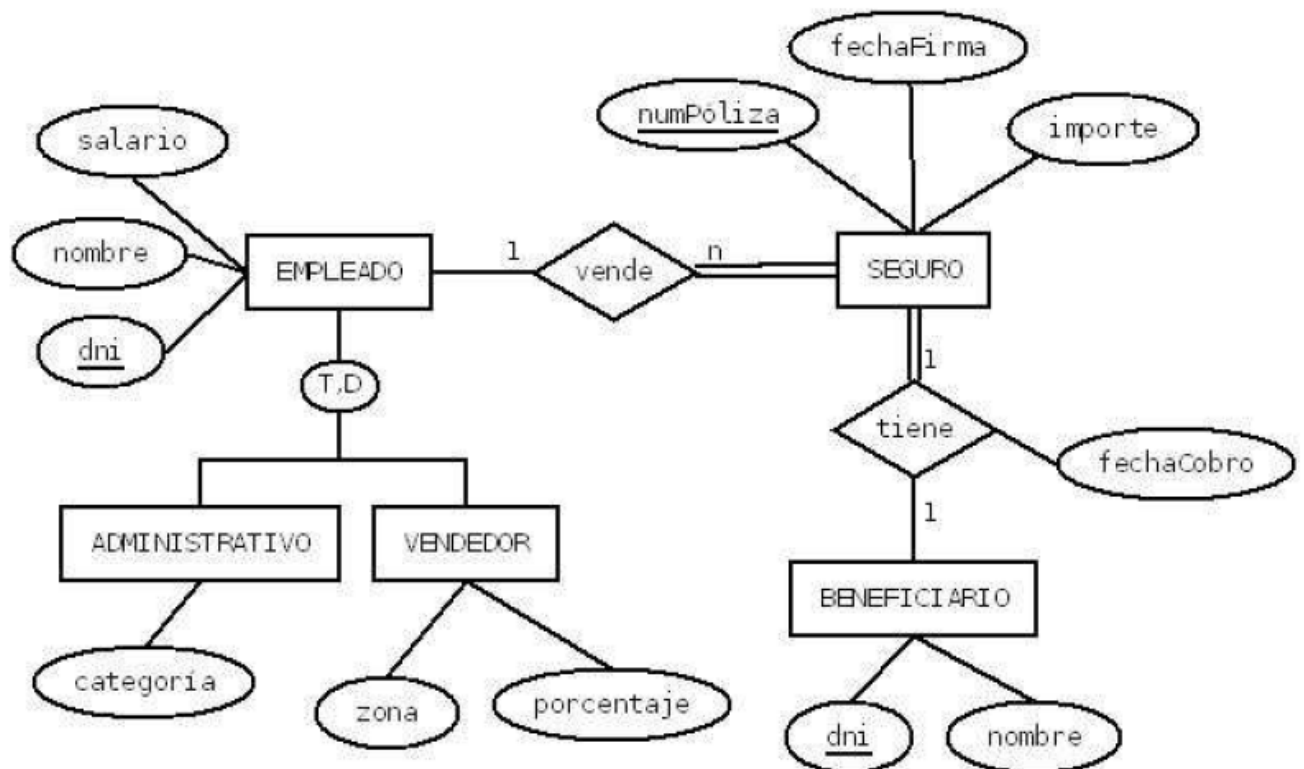


Compañía de seguros



Aclaración: en nuestra aseguradora realizamos seguros de vida individuales, es decir, de un solo beneficiario y que pueden cobrar de una sola vez.

1) Transformamos las **entidades fuertes** (aquellas entidades que no dependen de otras para conformar su clave principal) y sus atributos.

EMPLEADO (dni, nombre, salario)
CP {dni}

SEGURO (numPoliza, fechaFirma, importe)
CP {numPoliza}

BENEFICIARIO (dni, nombre)
CP {dni}

2) Transformamos las **entidades débiles** y sus atributos. En este caso no hay ninguna entidad débil.

Sí hay una entidad (SEGURO) con restricciones de existencia con respecto a dos relaciones distintas pero ya vimos que eso no era lo mismo.

3) Transformamos las **especializaciones** de entidades.

Existe una jerarquía de EMPLEADOS en el E/R. Las entidades especializadas son ADMINISTRATIVO y VENDEDOR, así que, todas las ocurrencias de ADMINISTRATIVOS y VENEDORES serán también de tipo EMPLEADO.

Recuerda que las ocurrencias de ADMINISTRATIVO y VENDEDOR tienen los atributos comunes de EMPLEADO y, además, los suyos propios.

La relación (relation) EMPLEADO del esquema lógico relacional no sufre ninguna transformación por tener especializaciones pero la reescribo aquí para que resulte más clara la transformación de sus especializaciones.

```
EMPLEADO (dni, nombre, salario)
CP {dni}
```

```
ADMINISTRATIVO (dni, categoria)
CP {dni}
CAjena: {dni} -> EMPLEADO.dni
```

Como todos los ADMINISTRATIVOS son también EMPLEADOS utilizaremos la misma CP. Y nos aseguraremos de que los usuarios de la BD no puedan insertar un DNI de ADMINISTRATIVO si no es también un DNI de EMPLEADO definiéndolo como clave ajena.

```
VENDEDOR (dni, zona, porcentaje)
CP {dni}
CAjena: {dni} -> EMPLEADO.dni
```

¿Qué pasaría si hubiésemos definido la clave ajena de VENDEDOR de esta forma:
CAjena: {dni} → ADMINISTRATIVO.dni? Que estaríamos prohibiendo que existiese un DNI en VENDEDOR que no fuese a su vez ADMINISTRATIVO. Habríamos cambiado la jerarquía de las entidades.

Ahora debemos fijarnos en el **tipo de especialización**:

- Total: Debemos obligar a que todos los DNIs de EMPLEADO aparezcan en ADMINISTRATIVO o en VENDEDOR. Dicho de otro modo:

RI1: No puede existir un valor en EMPLEADO.dni que no aparezca en ADMINISTRATIVO.dni o en VENDEDOR.dni.

- Disjunta: Debemos obligar a que no exista un EMPLEADO que pueda ser a la vez ADMINISTRATIVO y VENDEDOR. Dicho de otro modo:

RI2: Un valor de ADMINISTRATIVO.dni no puede aparecer en VENDEDOR.dni y viceversa.

4) Transformamos las **relaciones** y sus atributos.

No es necesario seguir ningún orden a la hora de seleccionar qué relaciones transformar primero pero yo recomiendo empezar por las que te resulten más sencillas.

a) Relación binaria VENDE 1:N



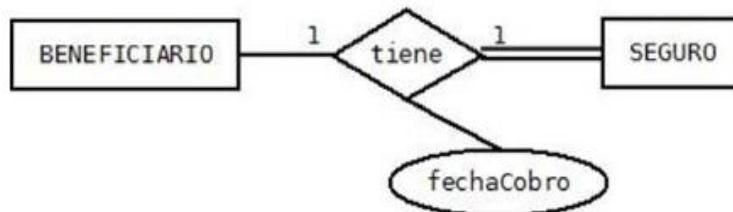
Las relaciones binarias 1:N del E/R se transformaban añadiendo una clave ajena a la entidad que tenía cardinalidad máxima igual a 1. En este caso es la entidad SEGURO la que contendrá una clave ajena a la entidad EMPLEADO.

```
SEGURO (numPoliza, fechaFirma, importe, dni_empleado)
CP: {numPoliza}
CAjena: {dni_empleado} -> EMPLEADO.dni
```

Y nos falta un paso más porque aún no hemos controlado la RE. Debemos asegurarnos de que todos los SEGUROS de la BD tengan informado el EMPLEADO que lo ha vendido. Eso es tan sencillo como obligar a que el atributo SEGURO.dni_empleado esté siempre informado.

```
SEGURO (numPoliza, fechaFirma, importe, dni_empleado)
CP:{numPoliza}
CAjena: {dni_empleado} -> EMPLEADO.dni
VNN: {dni_empleado}
```

b) Relación binaria TIENE 1:1



Las relaciones binarias 1:1 del E/R sin restricciones de existencia (RE) tenían soluciones simétricas pero si alguna entidad participante tenía RE entonces era preferible que esa entidad fuese la que contuviese la clave ajena.

En este caso es la entidad SEGURO la que contendrá una clave ajena a la entidad BENEFICIARIO.

```
SEGURO (numPoliza, fechaFirma, importe, dni_empleado,
dni_beneficiario)
CP: {numPoliza}
CAjena: {dni_empleado}->EMPLEADO.dni
VNN: {dni_empleado}
CAjena: {dni_beneficiario}->BENEFICIARIO.dni
UNICO: {dni_beneficiario}
```

Y para captar la RE de SEGURO con respecto a la relación TIENE deberemos obligar a que todos los SEGUROS tengan como mínimo un beneficiario.

```
SEGURO (numPoliza, fechaFirma, importe, dni_empleado,
dni_beneficiario)
CP: {numPoliza}
CAjena: {dni_empleado}->EMPLEADO.dni
VNN: {dni_empleado}
CAjena: {dni_beneficiario}->BENEFICIARIO.dni
UNICO: {dni_beneficiario}
VNN: {dni_beneficiario}
```

Como la relación TIENE tiene un atributo en el E/R, debemos añadirlo a la relación del esquema lógico relacional que contiene la clave ajena.

```
SEGURO (numPoliza, fechaFirma, importe, dni_empleado,  
dni_beneficiario, fechaCobro)  
CP: {numPoliza}  
CAjena: {dni_empleado} -> EMPLEADO.dni  
CAjena: {dni_beneficiario} -> BENEFICIARIO.dni  
UNICO: {dni_beneficiario}  
VNN: {dni_empleado, dni_beneficiario}
```

Recuerda que las restricciones VNN se podían agrupar sin problemas.

En una situación normal, deberíamos añadir una RI que controlase que si el atributo SEGURO.dni_beneficiario es NULL también lo sea el atributo SEGURO.fechaCobro porque solo tiene sentido saber la fecha de cobro de un seguro por parte del beneficiario... cuando conocemos el beneficiario.

En este caso no será necesario, ¿ves el por qué? Al existir la restricción de VNN sobre el atributo SEGURO.dni_beneficiario, todos los SEGUROS tendrán el BENEFICIARIO informado, así que en todos los SEGUROS podremos informar la SEGURO.fechaCobro. Dicho de otro forma, no puede existir una tupla de SEGURO sin que participe en la relación con un BENEFICIARIO, así que siempre podremos informar el atributo fechaCobro de la relación.

El esquema lógico relacional final es el siguiente:

```
EMPLEADO (dni, nombre, salario)  
CP: {dni}  
  
ADMINISTRATIVO (dni, categoria)  
CP: {dni}  
CAjena: {dni} -> EMPLEADO.dni  
  
VENDEDOR (dni, zona, porcentaje)  
CP: {dni}  
CAjena: {dni} -> EMPLEADO.dni  
RI1: No puede existir un valor en EMPLEADO.dni que no aparezca en  
ADMINISTRATIVO.dni o en VENDEDOR.dni.  
RI2: Un valor de ADMINISTRATIVO.dni no puede aparecer en VENDEDOR.dni  
y viceversa.  
  
BENEFICIARIO (dni, nombre)  
CP: {dni}  
  
SEGURO (numPoliza, fechaFirma, importe, dni_empleado,  
dni_beneficiario, fechaCobro)  
CP: {numPoliza}  
CAjena: {dni_empleado} -> EMPLEADO.dni  
CAjena: {dni_beneficiario} -> BENEFICIARIO.dni  
UNICO: {dni_beneficiario}  
VNN: {dni_empleado, dni_beneficiario}
```