



En este tema vamos a hablar de la traducción, o mejor, la transformación de los conceptos representados en un esquema Entidad-Relación a sus correspondientes en Modelo Relacional. Esta "traducción", que se mostrará a continuación, siempre es igual estructuralmente, se trata de añadir claves ajenas en sitios concretos y modificar su comportamiento, en ocasiones, haciéndolas parte de la clave primaria, clave alternativa o simplemente de valor no nulo.

Es decir, se va a mostrar la equivalencia más eficiente entre las distintas relaciones representables en E-R y MR.

Este tema se está dando en paralelo a nuestra "segunda hora" donde, precisamente, ya se están mostrando esas estructuras en Modelo Relacional.

ER a MR

- Esquemas E-R generan esquemas MR en 3FN
 - las formas normales se estudian más adelante
- Pérdidas semánticas
 - no hay traducción
 - se elige la menos “mala”

El objetivo es transformar nuestro sistema de información descrito en Entidad-Relación a Modelo Relacional, a tablas, y poderlo manejar con un motor de base de datos de uso mayoritario (MySQL, por ejemplo).

La normalización es un tema que veremos más adelante, sin embargo adelantamos un aspecto ventajoso de diseñar con Entidad-Relación, y es que las tablas que obtendremos finalmente estarán en tercera forma normal (3FN), para entendernos, tablas eficientes desde el punto de vista estructural.

No todo lo que se expresa en E-R tiene traducción a MR, o al menos no directa. Estas "pérdidas semánticas" se dejan para solucionarlas más adelante, probablemente con programación de cliente o de servidor, o incluso se pierden en el proceso de desarrollo.

entidad a tabla

EMPLEADO
<u>dni</u>
población
nombre
nombre
apellidos
teléfono 0..N
formapago 1..1

EMPLEADO (dni, población,
nombre, apellidos, teléfono,
formapago)
CP(dni)
VNN(formapago)

El caso típico es la "traducción" de una entidad a una tabla. Los identificadores conforman la clave primaria, los atributos requeridos las definiciones de valor no nulo (columnas que no admiten nulos), en general cada atributo será una columna de esa tabla.

entidad a tabla: identificador compuesto

EMPLEADO	
<u>dni</u>	
<u>población</u>	
nombre	
nombre	
apellidos	
teléfono	0..N
formapago	1..1

EMPLEADO (dni, población, nombre, apellidos, teléfono, formapago)

CP(dni, población)

VNN(formapago)

Entiéndase que todo aquello que aparezca subrayado entrará a formar parte de la clave primaria, es el identificador, compuesto o no, de la entidad. Dicho de otra forma, NO terminan siendo dos claves candidatas distintas. En este ejemplo se parte de una entidad con un identificador compuesto (aunque no tenga mucho sentido como tal) para mostrar, precisamente, que se combinan en la clave primaria de la tabla resultado.

entidad a tabla: pérdidas

EMPLEADO	
<u>dni</u>	
población	
nombre	
nombre	
apellidos	
teléfono	0..N
formapago	1..1

EMPLEADO (dni, población,
nombre, apellidos, teléfono,
formapago)
CP(dni)
VNN(formapago)

Los atributos compuestos es una de esas pérdidas semánticas, aunque poco importante. Al final, simplemente tendremos una columna por cada atributo simple.

entidad a tabla: pérdidas

EMPLEADO
<u>dni</u>
población
nombre
nombre
apellidos
teléfono 0..N
formapago 1..1

EMPLEADO (dni, población,
nombre, apellidos, **teléfono**,
formapago)

CP(dni)

VNN(formapago)

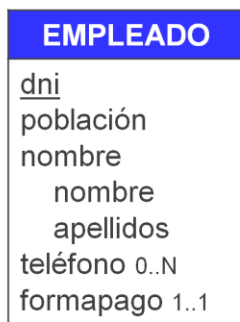
- hay alternativas
 - EMPLEADO (dni, población, nombre, apellidos, formapago) CP (dni) VNN(formapago)
 - TELÉFONOS (dni, teléfono) CP(dni,teléfono) CAj(dni)→EMPLEADO



Los atributos multivaluados no tienen una traducción directa, hay que tomar alguna decisión de diseño y generar alguna tabla adicional. Aquí se muestra una de esas soluciones posibles.

En el peor de los casos se puede decidir, simplemente, que no se mantiene ese carácter de multivaluado y se deja como una columna más. También, en muchas ocasiones se opta por tener 2 o 3 columnas para teléfonos (teléfono1, teléfono2, teléfono3, por ejemplo). En todo caso, el diseñador de la base de datos debe decidir qué es mejor para un correcto balance entre restricciones deseadas y operativa de la base de datos en producción.

relaciones a tabla: 1:N



EMPRESA (cif, nombre, país_sede)
CP(cif)

EMPLEADO (dni, población,
nombre, apellidos, teléfono,
formapago, **empresa**)

CP(dni)

VNN(formapago)

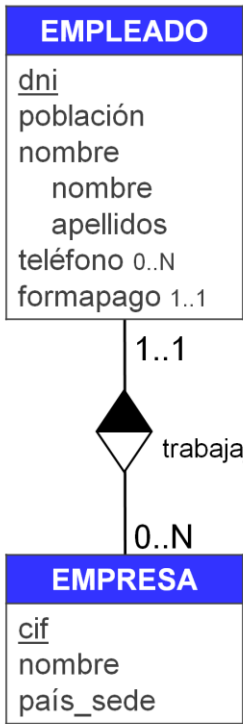
CAj (empresa) → EMPRESA

En cuanto a las correspondencias entre clases, las relaciones entre entidades, todo está muy ligado a lo que se está viendo relativo al modelo relacional. Evidentemente, si el E-R muestra una relación 1:N su traducción a tablas será una relación 1:N... en tablas.

En cualquier caso, las relaciones en E-R, las cardinalidades y los triángulos, se traducen a una clave ajena, es decir, una (o varias, depende de la clave primaria a la que apunte) columnas adicional que es clave ajena. Siempre se trata de columnas que hay que añadir a la tabla resultado.

Por eso no hace falta especificarlas en E-R, lo importante es el hecho que representa, que hay una relación entre el concepto A y el B. En E-R se hace dibujando los símbolos correspondientes; en MR añadiendo claves ajenas.

relaciones a tabla: 1:N+RE



EMPRESA (cif, nombre, país_sede)
CP(cif)

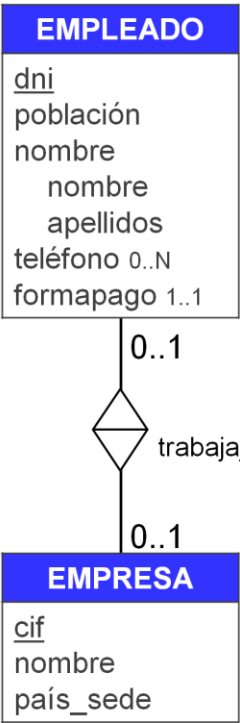
EMPLEADO (dni, población,
nombre, apellidos, teléfono,
formapago, **empresa**)

CP(dni)
VNN(formapago)

CAj (empresa) → EMPRESA
VNN (empresa)

En el caso de las relaciones uno a muchos (1:N) se trata de colocar una clave ajena en la tabla que sufre las restricciones. En este caso, es el empleado el que no puede trabajar en 2 empresas simultáneamente. Además, como se le exige trabajar en al menos una, siempre, por eso el valor no nulo que se añade a la clave ajena, para que siempre tenga un valor; además, ese valor debe cumplir con la integridad referencial.

relaciones a tabla: 1:1



EMPRESA (cif, nombre, país_sede) CP(cif)

EMPLEADO (dni, población, nombre, apellidos, teléfono, formapago)

CP(dni) VNN(formapago)

TRABAJA_EN (empleado, empresa)

CP(empleado) Calt (empresa)

CAj (empleado) → EMPLEADO

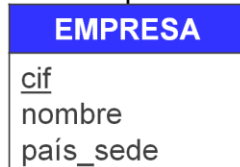
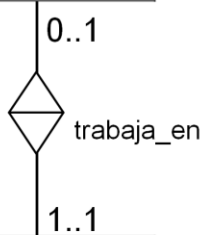
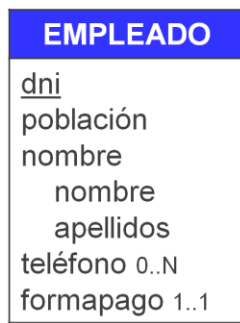
CAj (empresa) → EMPRESA



En otros casos, la solución pasa por definir una tercera tabla. Las relaciones 1:1 sin restricciones de existencia se deben representar así. Ahora necesitamos 2 claves ajenas porque ni EMPLEADO ni EMPRESA van a albergar esa referencia al otro, las referencias están ahora en otra tabla.

Da igual qué clave ajena se elije como primaria y cuál como clave alternativa.

relaciones a tabla: 1:1+RE



EMPRESA (cif, nombre, país_sede,
empleado)

CP(cif)

CAIt (empleado)

CAj (empleado) → EMPLEADO

EMPLEADO (dni, población, nombre, apellidos,
teléfono, formapago)

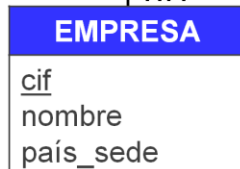
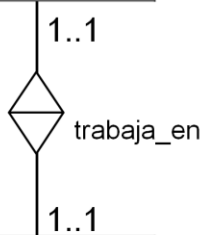
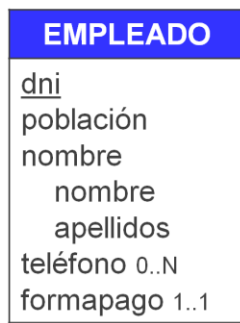
CP(dni) VNN(formapago)



Sin embargo, la mejor forma de traducir una relación 1:1 con restricción de existencia en una de las entidades es añadir una clave ajena en una de las tablas, la que tenga esa restricción de existencia, ya que es la única forma que tenemos de obligar a "alguien" a relacionarse con "otros". Dicho de otra forma, nunca podremos representar una restricción de existencia en una tabla si la clave ajena de la relación no está en la propia tabla.

En este caso, como además el máximo de la relación es "1", volvemos hacer uso de la clave alternativa, que ahora será la clave ajena que hemos de añadir a EMPRESA.

relaciones a tabla: 1:1+2RE



TRABAJA_EN(cif, nombre,
país_sede, dni, población,
nombre, apellidos, teléfono,
formapago)

CP(cif)

CAlt(dni)

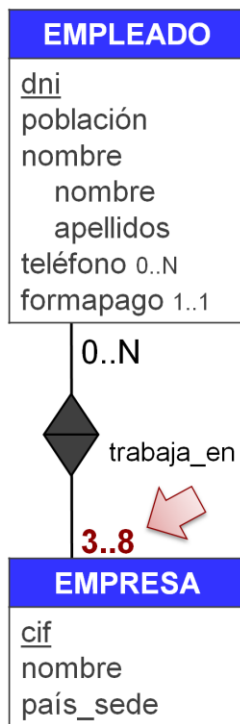
VNN(formapago)



Como ya hemos mencionado, la 1:1 con doble restricción de existencia es una relación "rara" (que termina casi siempre cambiando las pautas de diseño para transformarla, ya, en una única entidad). No obstante, teniendo cierto sentido representarla así en E-R, hay que tener claro que su traducción, la verdaderamente eficiente desde un punto de vista estructural, es una única tabla sin claves ajenas pero con una clave primaria y otra alternativa.



relaciones a tabla: pérdidas



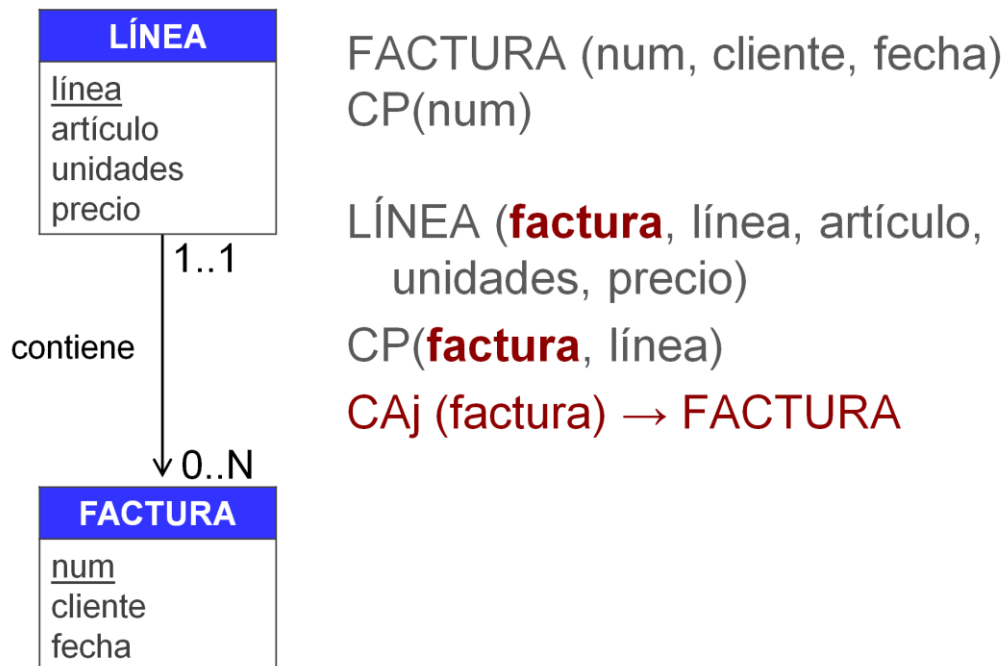
EMPRESA (cif, nombre, país_sede)
CP(cif)

EMPLEADO (dni, población, nombre, apellidos, teléfono, formapago)
CP(dni) VNN(formapago)

TRABAJA_EN (empleado, empresa)
CP(empleado, empresa)
CAj (empleado) → EMPLEADO
CAj (empresa) → EMPRESA

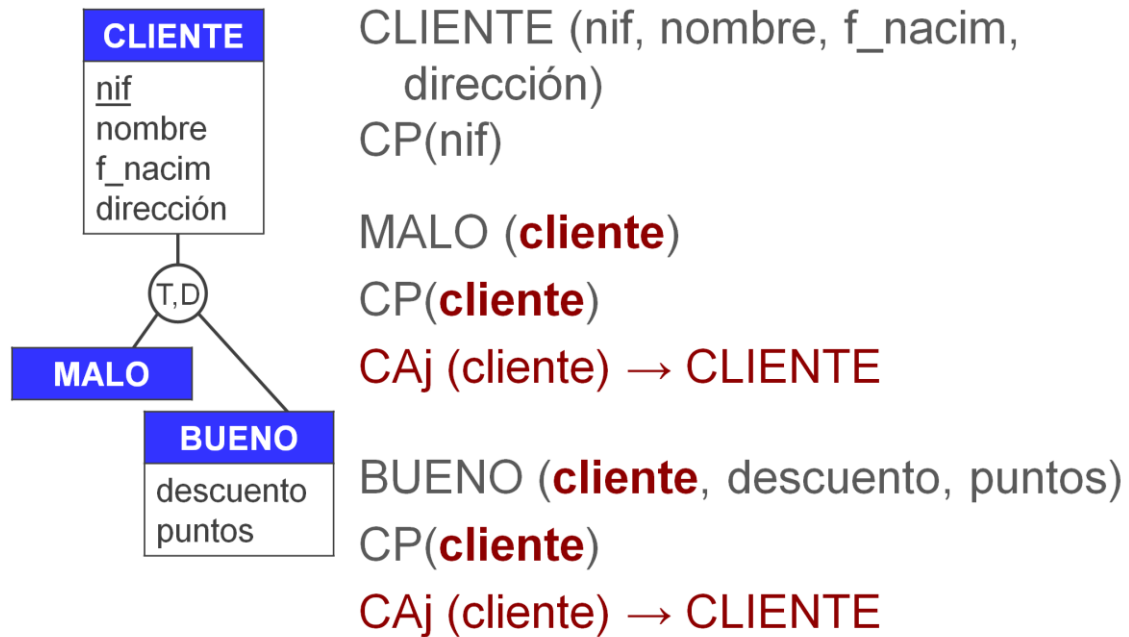
En realidad, cualquier cardinalidad que no sea 0,1 o N se ha de considerar una pérdida semántica que podrá ser resuelta o paliada, o no, más adelante.

relaciones a tabla: dep. ID



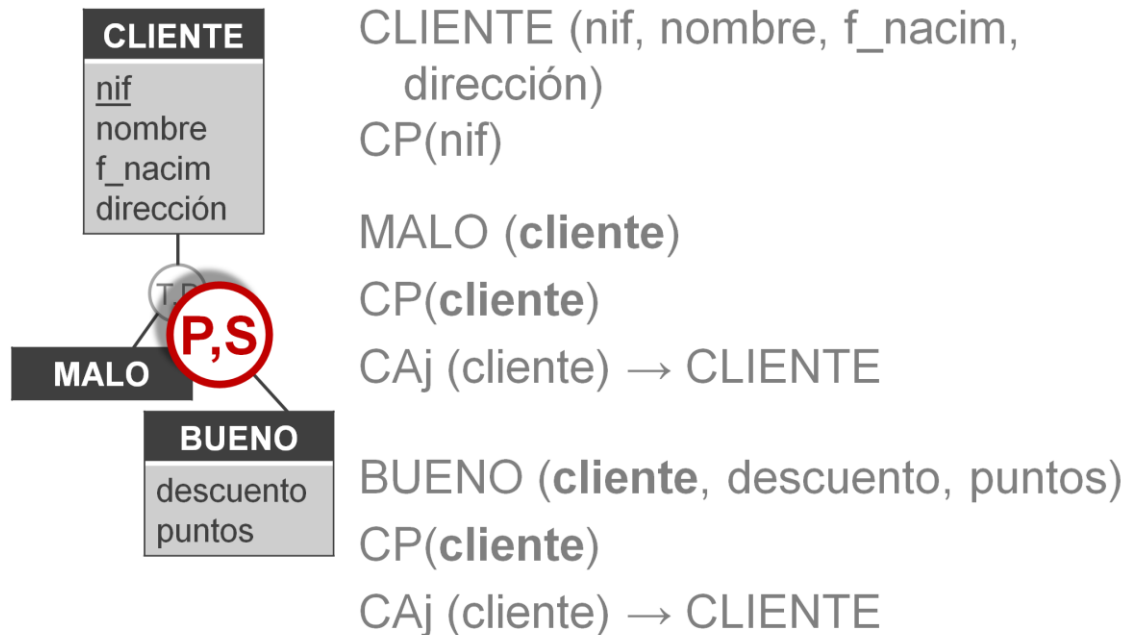
El caso de la dependencia de identificador, típica de los documentos tipo factura, se traduce a tablas como indica el nombre del tipo de correspondencia: LÍNEA necesita apoyarse en el identificador de FACTURA para construir su propio identificador, es decir, debemos añadir una clave ajena e incorporarla a la clave primaria.

relaciones a tabla: generalización



En el caso de las generalizaciones, aparte de la tabla con los datos comunes a todas las subclases, necesitamos una tabla por cada categoría que nos encontremos en el E-R. Esas tablas comparten el hecho de que definen una clave ajena que es exactamente la clave primaria. Aparte, se definirán tantas columnas como atributos propios tenga la entidad.

relaciones a tabla: generalización: pérdidas



En cuanto a las propiedades de cobertura, pongamos lo que pongamos en el E-R, necesariamente se ha de transformar en "parcial y solapada" en modelo relacional. Es la única combinación que admite el modelo, otra pérdida semántica en el paso de E-R a MR. No obstante, esto se puede controlar, por ejemplo, mediante disparadores (triggers), un tipo especial de procedimientos programados en el servidor de base de datos.

conclusión

- Hemos dado por supuesto dominios-tipos de datos
- A partir de estas transformaciones básicas se genera un esquema MR completo
- La mayoría de las veces no hay ambigüedad en la transformación
 - la “famosa” 3FN
- Las “pérdidas” se pueden solventar por otros medios