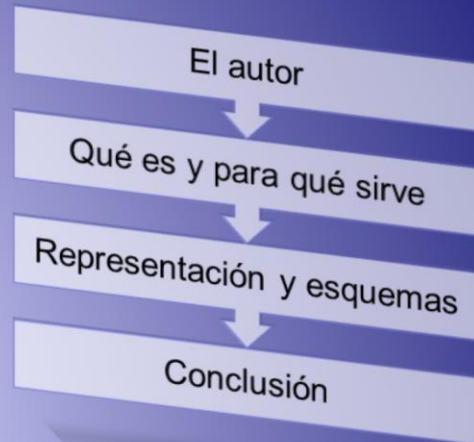


entidad-relación  
Tema 2B



## el autor



- Peter Chen

- Dr. **Peter Pin-Shan Chen**
- Peter P. Chen. *The Entity-Relationship Model: Toward a Unified View of Data Export*. ACM Transactions on Database Systems, Vol. 1 (1976), pp. 9-36.
- wikipedia



El Dr. **Peter Pin-Shan Chen** es el creador del Modelo Entidad-Relación (Modelo ER). En el año 1968, obtuvo el grado de Licenciado en Ciencias en la Universidad Nacional de Taiwán. Posteriormente, en el año 1973, obtuvo el grado académico de Doctor en Ciencias de la Computación y Matemáticas Aplicadas en la Universidad de Harvard. Desde 1983, el Dr. Peter Chen disfruta del cargo de M. J. Distinguished Chair Professor of Computer Science en la Universidad del Estado de Louisiana.

El artículo original de Peter Chen[1] sobre el Modelo ER es uno de los trabajos más citados en el campo de las ciencias de la computación. Este artículo ha sido recientemente galardonado como uno de los 38 artículos más influyentes para las ciencias de la computación, según una encuesta realizada a cerca de 1000 investigadores universitarios.

[1] Peter P. Chen. *The Entity-Relationship Model: Toward a Unified View of Data Export*. ACM Transactions on Database Systems, Vol. 1 (1976), pp. 9-36.

## qué es y para qué sirve

- Modelo de datos semántico gráfico
  - Se supone de la tercera generación de MM.DD.
- Para representar sistemas de información
  - Históricamente, el primer paso en el diseño de una BD
  - Esquema conceptual
    - Visión de alto nivel, sin excesivos detalles de su futura implementación
  - Propiedades estáticas del S.I.
    - No detalles de procesos o transacciones, solo estructura



El E-R (y posteriormente el EER, Entidad-Relación Extendido) es la frontera entre los modelos de datos clásicos y semánticos. Aún cuando tiene alguna herramienta de representación más que el modelo relacional, son muy cercanos los dos. Hay que decir, también, que no todo lo que se puede expresar en modelo relacional tiene su correspondencia en E-R, es decir, aunque asumimos que el modelo E-R es posterior y algo más rico expresivamente, las cosas que podemos "decir" usando el modelo relacional NO es un subconjunto exacto del E-R.

En cualquier caso, no se resuelve la falta de conceptos para expresar restricciones dinámicas, las que tiene que ver con los procesos en los que usan los datos, la parte estática: cómo se genera una factura, cómo se realiza un pedido a proveedores, etc.

## qué es y para qué sirve

- No existen SGBD basados en E-R
  - ¿Entonces?
    - Traducible fácilmente a MR y directamente en tercera forma normal
- **AVISO**
  - nuestra notación no es estándar
  - mezcla de UML y clásica
  - lo importante son las estructuras del modelo y los conceptos a representar



No existen SGBD... que hayan tenido éxito, alguna propuesta hubo.

El concepto de forma normal se verá en un tema posterior pero, para entendernos, el E-R consigue "automatizar" un paso de diseño que trabajando únicamente con modelo relacional es casi obligador realizar hasta obtener una estructura de datos eficiente. Por contra, esta "automatización" hace que "perdamos" algún detalle que el modelo relacional sí es capaz de representar; no hay peligro, son muy pocos estos casos y, en cualquier caso, para eso está el diseño lógico y físico.

Notaciones hay muchas, unas más comunes que otras. En nuestro caso, hemos decidido adoptar una variante definida por nosotros, sin detrimento de que puede transformarse sin pérdidas a cualquier otro sistema de símbolos E-R.

## representación: entidad

- Concepto, clase de objetos
  - Una entidad se describe por sus atributos
    - Identificador
    - Compuesto
    - Multivaluado
    - No nulo
    - ...hay más
      - derivado
      - identificador alternativo
      - ...

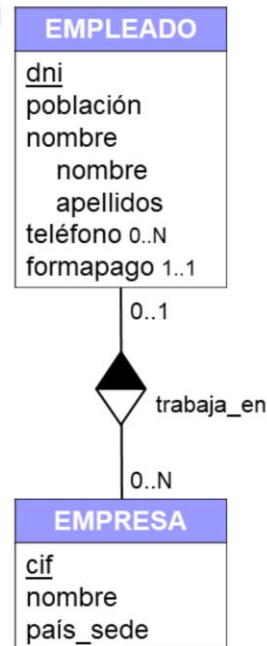
EMPLEADO
<u>dni</u>
población
nombre
nombre
apellidos
teléfono 0..N
formapago 1..1



Las entidades son el primer paso, los ladrillos con los que vamos a construir todo nuestro sistema de información.

## representación: relación

- Indica una acción o relación entre 2 entidades
  - Los empleados **pertenecen** a empresas
  - Cada empleado pertenece a ninguna o, como mucho, 1 empresa
    - los empleados **pueden** pertenecer a 1 empresa
  - Cada empresa tiene ningún o muchos empleados
    - las empresas **pueden** tener muchos empleados
      - no hay límite máximo



Una vez que ya tenemos entidades, podemos enriquecer el esquema con las relaciones que hay entre ellas.

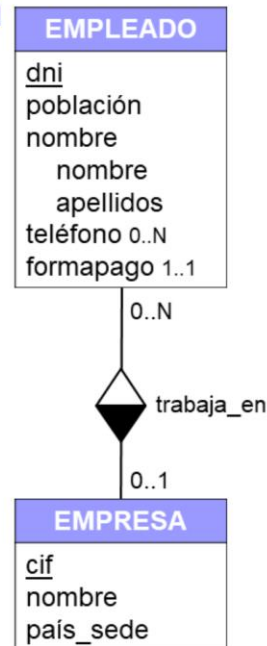
Téngase en cuenta que el diseño del esquema de una base de datos es una tarea compleja que está más allá de los objetivos de la asignatura. Hablamos de decenas, cientos de entidades, y tantas o muchas más relaciones entre ellas, como escenario normal de un diseñador de bases de datos. Aquí solo estamos mostrando los fundamentos del modelo y sistemas de información extremadamente simples.

Al definir relaciones, estamos estableciendo ciertos límites a la hora de insertar datos en nuestra base de datos. Cosas como que si permitimos que un empleado trabaje en más de una empresa simultáneamente o no. Son decisiones que toma el diseñador en función de cómo quiere que se comporte el operador que maneja los datos a almacenar.

## representación: relación

- Indica una acción o relación entre 2 entidades

- Los empleados **pertenecen** a empresas
- Cada empleado pertenece a ninguna o **muchas** empresas
  - los empleados **pueden** pertenecer a muchas empresas
- Cada empresa tiene ninguno o **1** empleado
  - las empresas **pueden** tener 1 empleado



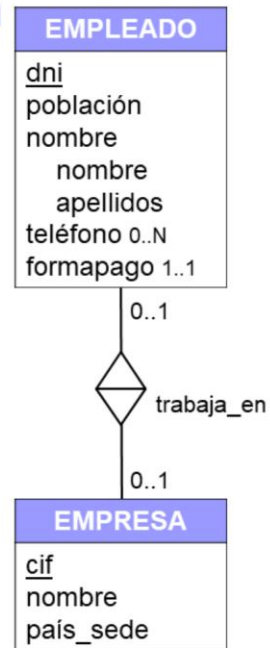
Los triángulos, rellenos o vacíos, son redundante, los utilizamos para que, visualmente, sea más fácil identificar el tipo de cardinalidad que se está representando. Por ejemplo, y en este caso, el EMPLEADO tiene unos valores mínimo y máximo de (0,N), una relación "uno a muchos" con EMPRESA, que se corresponde con el triángulo relleno opuesto.

De hecho, los triángulos hacen referencia a la cardinalidad máxima únicamente (el número a la derecha: x..N o x..1). Los valores mínimos, "1" como se verá más adelante, solo se pueden definir en las expresiones numéricas que acompañan a la relación y la entidad.

## representación: relación

- Indica una acción o relación entre 2 entidades

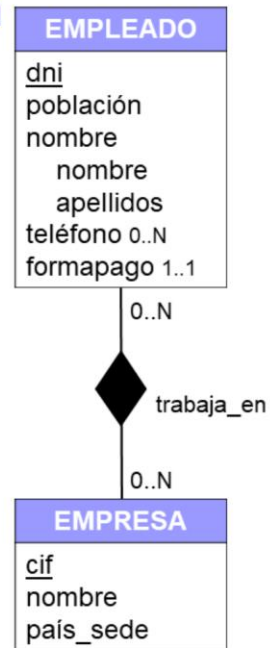
- Los empleados **pertenecen** a empresas
- Cada empleado pertenece a ninguna o **1** empresa
  - los empleados **pueden** pertenecer a 1 empresas
- Cada empresa tiene ninguno o **1** empleado
  - las empresas **pueden** tener 1 empleado





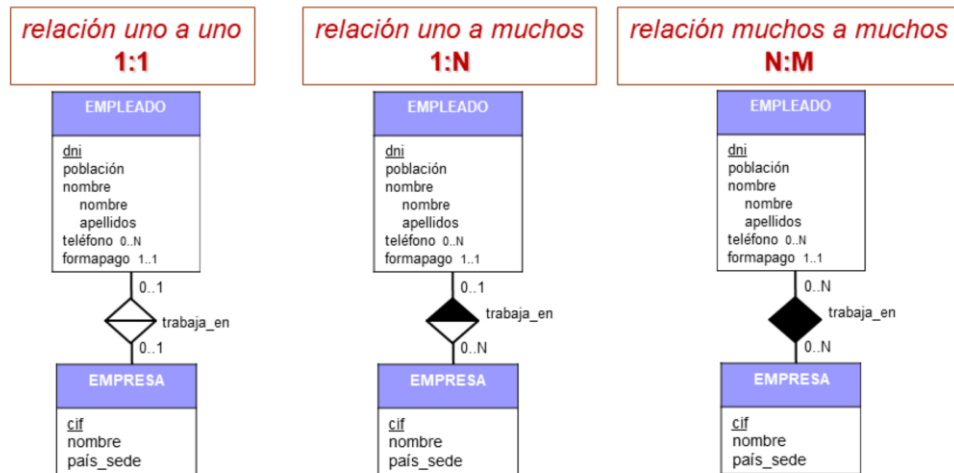
## representación: relación

- Indica una acción o relación entre 2 entidades
  - Los empleados **pertenecen** a empresas
  - Cada empleado pertenece a ninguna o **muchas** empresas
    - los empleados **pueden** pertenecer a muchas empresas
  - Cada empresa tiene ninguno o **muchos** empleados
    - las empresas **pueden** tener muchos empleado



## representación: relación

- Terminología



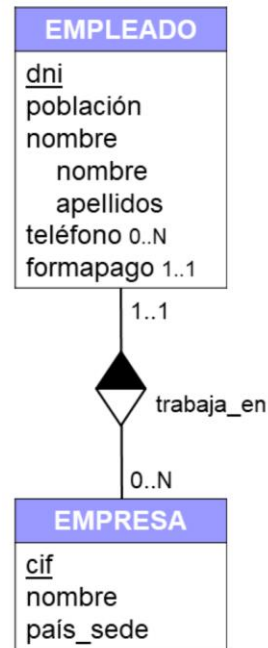
Casi diríamos que "coloquialmente" se suele hablar de relaciones "uno a uno", "uno a muchos" y "muchos a muchos". En realidad, falta todavía hablar de restricciones de existencia y, en el caso de las 1:N, quién es el "uno" y quién el "muchos".

Pero es perfectamente válido para describir lo general de la relación.

## representación: restricciones

- Existencia

- obligación de estar relacionado con **al menos 1** objeto de la otra relación
- los empleados **deben** pertenecer al menos a 1 empresa y **solo** a una empresa
- las empresas **pueden** tener muchos empleados
  - se sobreentiende que pueden **no** tener

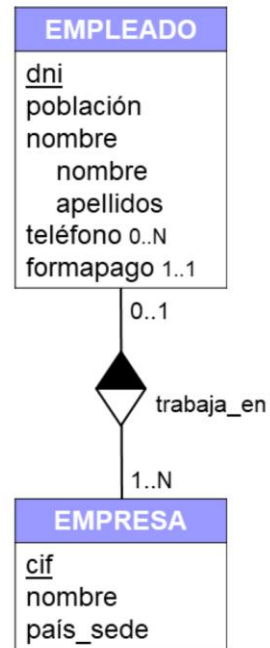


Las restricciones de existencia indican que un elemento de la base de datos NO puede existir, no puede ser almacenado en nuestra base de datos si no está relacionado con otro individuo de la otra entidad. Aquí, que los empleados, al ser dados de alta en nuestro sistema, obligatoriamente hay que decir en qué empresa trabajan. Si no es así, no se puede almacenar al empleado en la base de datos.

## representación: restricciones

- Existencia

- obligación de estar relacionado con **al menos 1** objeto de la otra relación
- los empleados **pueden** pertenecer a, como mucho, 1 empresa
- las empresas **deben** tener empleados
  - al menos uno**
  - se sobreentiende que pueden tener muchos, no se dice nada que lo limite
    - no es demasiado útil en cardinalidades ...a muchos

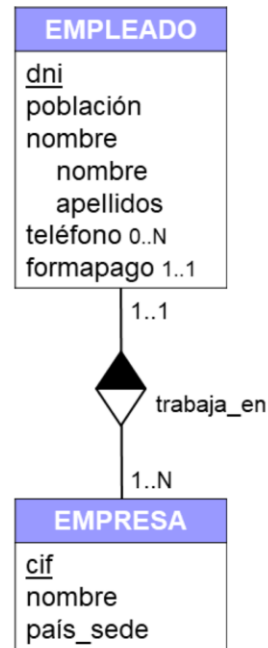


Mucho cuidado. Esta es una de las cosas que Sí se pueden decir en E-R y NO se pueden traducir luego a modelo relacional.

## representación: restricciones

- Existencia

- obligación de estar relacionado con **al menos 1** objeto de la otra relación
- los empleados **deben** pertenecer a 1 y solo 1 empresa
- las empresas **deben** tener empleados
- muy discutible la necesidad de estas cosas...

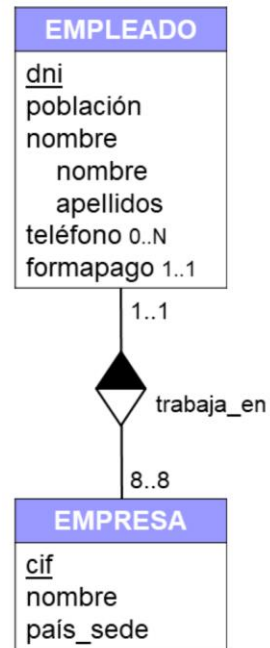


Hay que tener mucho cuidado con lo que se diseña. Aquí estamos estableciendo una situación que tiene una difícil solución en la base de datos final, típicamente relacional. Si el empleado necesita que exista previamente una empresa en la que va a trabajar, y la empresa necesita que exista al menos un empleado que asignarle, ¿por dónde empezamos a insertar datos en la BD? Habría que implementar un proceso especial de inserción de datos para este caso.

En general, hay que poner única y exclusivamente las restricciones de existencia que consideremos imprescindibles.

## representación: restricciones

- No hay limitaciones
  - las empresas **deben** tener 8 empleados
  - pero tendremos problemas cuando queramos pasarlo a modelo relacional

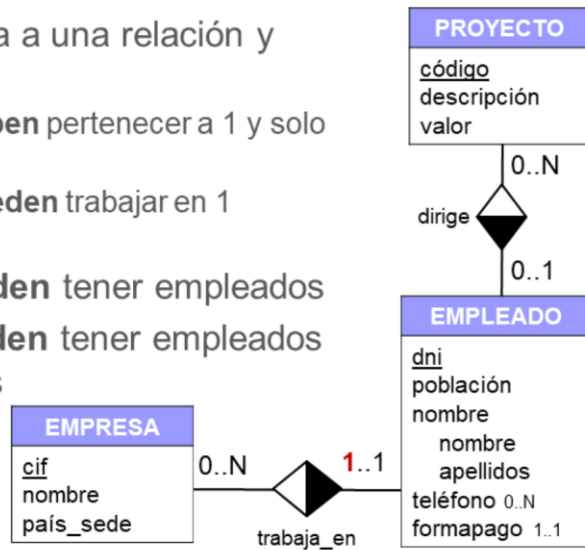


Esta es otra cosa que sí se puede decir en E-R pero no en modelo relacional. Posteriormente, seguramente mediante el programa que accede a la base de datos, se podría implementar el proceso que verifique esta restricción tan especial.

## representación: restricciones

- Existencia

- la restricción afecta a una relación y una entidad
  - los empleados **deben** pertenecer a 1 y solo 1 empresa
  - los empleados **pueden** trabajar en 1 proyecto
- las empresas **pueden** tener empleados
- los proyectos **pueden** tener empleados trabajando en ellos



Hasta ahora no habíamos mostrado entidades trabajando para varias relaciones. En todo caso, cada relación tiene sus propias restricciones independientemente de las demás.

## representación: restricciones

- Existencia

- la restricción afecta a una relación y una entidad
  - los empleados **deben** pertenecer a 1 y solo 1 empresa
  - los empleados **deben** trabajar en 1 proyecto y solo en uno
- las empresas pueden tener empleados
- los proyectos pueden tener empleados trabajando en ellos





## representación: restricciones

- Dependencia de identificador
  - el caso "Factura"

num	cliente	fecha
F001	ANTONIO	02/10/2009

línea	artículo	unidades	precio
L001	lápiz nº2	10	0.35
L002	borrador	3	0.10
L003	dinA4	100	0.05

num	cliente	fecha
F002	PEPE	05/10/2009

línea	artículo	unidades	precio
L001	sacaps	2	0.15
L002	pilot	300	1.10
L003	lápiz nº2	100	0.25
L004	carpeta	50	0.10

num	cliente	fecha
F003	ANTONIO	06/10/2009

línea	artículo	unidades	precio
L001	lápiz nº2	100	0.10

fundamentos de las bases de datos

Este es un caso bastante habitual en los sistemas de información administrativos. Los documentos tipo pedido, albarán, factura, etc. Son estructuras de datos que se componen de una cabecera con los datos generales y una serie de líneas de detalle que muestran, en este caso, los artículos que se están facturando.

Todas las facturas tienen varias líneas que comienzan a contar por la línea número 1. Así, tenemos un problema porque son con "línea 1" no tenemos suficiente información para distinguir entre líneas de detalle.

## representación: restricciones

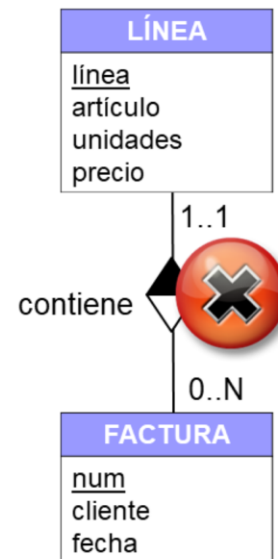
- Dependencia de identificador
  - el caso "Factura"

✓	num	cliente	fecha
	F001	ANTONIO	02/10/2009
	F002	PEPE	05/10/2009
	F003	ANTONIO	06/10/2009

✗	línea	artículo	unidades	precio
	L001	sacaps	2	0.15
	L002	pilot	300	1.10
	L003	lápiz nº2	100	0.25
	L004	carpeta	50	0.10
	L001	lápiz nº2	100	0.10
	L001	lápiz nº2		
	L002	borrador		
	L003	dinA4	100	0.05

información insuficiente en las líneas de detalle




18

Aquí vemos varias líneas "L001", por poner un caso. ¿Cómo puedo diferenciar entre ellas y consultar una en concreto?


## representación: restricciones

- Dependencia de identificador

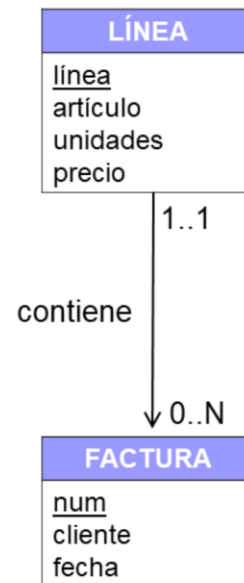
- una entidad necesita el identificador de otra
- solo en 1:N



num	cliente	fecha
F001	ANTONIO	02/10/2009
F002	PEPE	05/10/2009
F003	ANTONIO	06/10/2009



factura	línea	artículo	unidades	precio
F002	L001	sacaps	2	0.15
F002	L002	pilot	300	1.10
F002	L003	lápiz nº2	100	0.25
F002	L004	carpeta	50	0.10
F003	L001	lápiz nº2	100	0.10
F001	L001	lápiz nº2	10	0.35
F001	L002	borrador	3	0.10
F001	L003	dinA4	100	0.05



19

Se trata de un problema de diseño: la dependencia de identificador. Sí, es la línea L001 pero de la factura F001 o de la F002.

LÍNEA no dispone de información suficiente para construir un identificador válido y necesita agregar el identificador de otra entidad con la que se relacione, FACTURA en este caso.

En realidad, el identificador de cada línea es compuesto:

F001-L001,

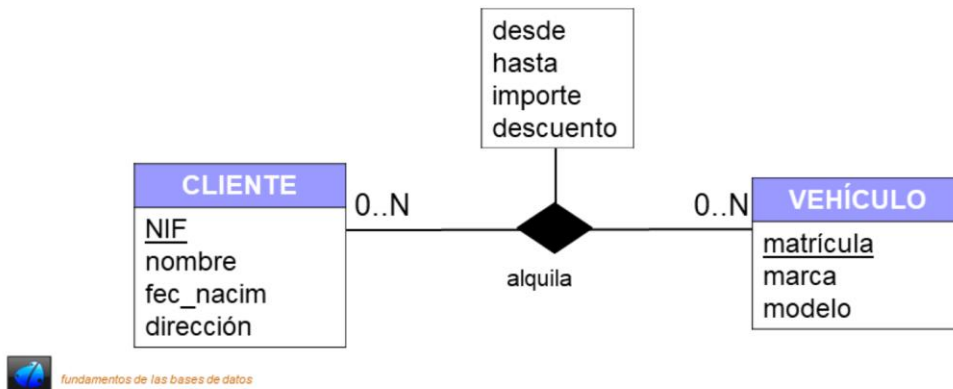
F001-L002,

F002-L001...

Todas son líneas diferentes.

## representación: relaciones y atributos

- Pueden pertenecer a relaciones
  - solo se “rellenan” cuando ocurre que un cliente alquila un vehículo

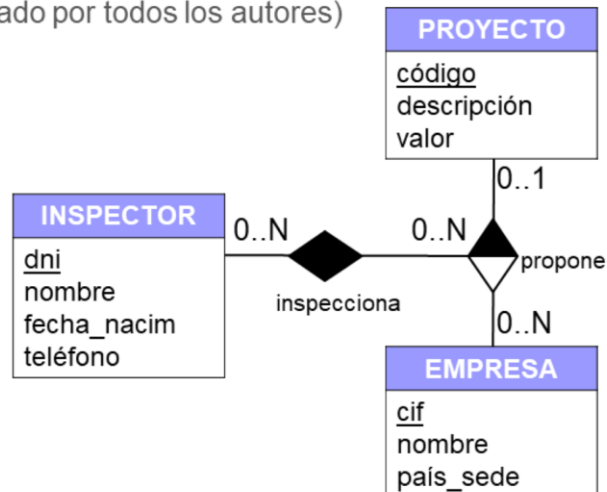


"Desde", "hasta", "importe" y "descuento" solo tienen sentido cuando acompañan a una pareja CLIENTE-VEHÍCULO; solo pueden existir como valores asociados a una relación entre un cliente y un vehículo.

## representación: agregación

- Entidades “complejas”

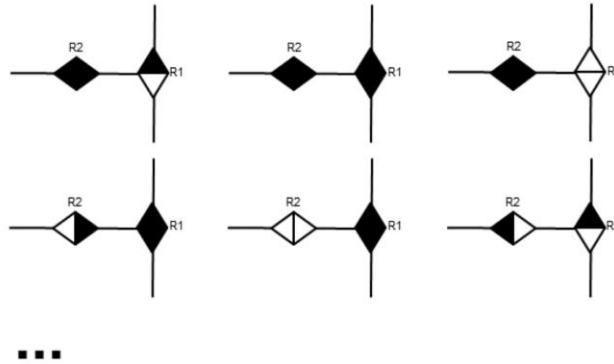
- los inspectores solo inspeccionan proyectos que son propuestos por alguna empresa
  - (no aceptado por todos los autores)



Esta "agregación" es, en realidad, un caso particular del concepto "agregación general" de todos los modelos de datos. Entiéndase que agregar es tanto definir una entidad por sus atributos como **construir una relación que tiene en alguno de sus extremos otra relación** que es, esto último, a lo que realmente nos referimos en E-R cuando hablamos de agregación.

## representación: agregación

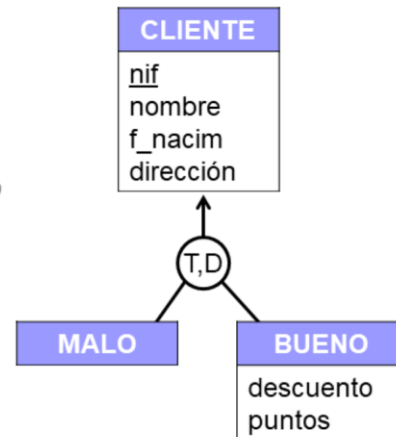
- Posibles representaciones
  - y muchas más
    - otra cosa es que sean útiles para casos concretos



Podemos agregar con las restricciones que nos apetezcan, dependerá del sistema de información que queramos representar.

## representación: generalización

- Tipos dentro de una entidad
  - hay clientes “buenos” y “malos”
  - los “buenos” pueden tener *descuento* y *puntos*; los “malos”, no
  - herencia
    - tanto “buenos” como “malos” tienen *nif*, *nombre*, *f\_nacim* y *dirección*
  - las propiedades de cobertura deben ser **Total o Parcial** y **Disjunta o Solapada**
    - aquí, todos los clientes son de algún tipo (total) pero ninguno es “bueno” y “malo” al mismo tiempo (disjunta)



Es exactamente el mismo concepto visto en el tema "Modelos de datos".

## conclusión

- E-R es un modelo de datos
  - acercamiento a *Orientación a Objetos*
- Estructuras
  - entidad, atributo y relación
- La primera definición de Chen ha tenido varias extensiones
  - E-R Extendido (EER)
    - más conceptos de representación (*generalización*, p. ej.)
- No hay SGBD basado en él
  - de uso generalizado, al menos





## conclusión

- Implementa los mecanismos de abstracción comunes
  - *clasificación, agregación (general), generalización*
  - restricciones de *dominio* (no las hemos utilizado), *identificación* y de *correspondencia* entre clases
- Muchas notaciones
  - no hay exactamente un estándar
  - nuestra notación, menos estándar todavía
- Muchas herramientas CASE
  - aunque muchas son meros "tableros de dibujo"
  - otras incorporan "traducción" a modelo relacional



Las restricciones de dominio son aquellas que definen que valores podemos utilizar en cada atributo. En una base de datos implementada en un SGBD relacional se convierte en el tipo de datos de cada columna definida.

CASE = *Computer Aided Software Engineering*

## conclusión: referencias

- [fbddocs.dlsi.ua.es/lecturas](http://fbddocs.dlsi.ua.es/lecturas)

