

MODELO RELACIONAL

Bases de datos (GIITIN)

Grado en Ingeniería Informática en Tecnologías de la Información

Elena Montañés Roces

Índice

- Modelo Relacional
- M. Entidad-relación/Relacional
- Restricciones inherentes y semánticas
- Conversión

Modelo Relacional

- Una **base de datos relacional** consiste en un conjunto de relaciones que sirve para representar tanto los objetos como las asociaciones entre ellos.
- Un **esquema de relación** se compone de:
 - un nombre de relación R
 - un conjunto de atributos $\{A_i\}$
 - un conjunto de n dominios (no necesariamente distintos) $\{D_i\}$ donde cada atributo estará definido sobre un dominio:

$$R(A_1(D_1), \dots, A_n(D_n))$$

Modelo Relacional

- Una **relación** es un conjunto de tuplas $\{t_k\}$:
 - $r(R) = \{t_k = (\langle A_1: v_{1k} \rangle, \dots, \langle A_n: v_{nk} \rangle) : v_{ik} \in D_i\}$
- Una **tupla** (t_k) es un conjunto de pares $\{\langle A_1: v_{1k} \rangle, \dots, \langle A_n: v_{nk} \rangle\}$ donde:
 - A_i es el nombre de un atributo $i=1, \dots, n$
 - v_{ik} es un valor del correspondiente dominio D_i sobre el que está definido el atributo A_i .
- El número de atributos de una relación se denomina **grado** de la relación.
- El número de tuplas de una relación se denomina **cardinalidad** de la relación.

Ejemplo

□ Esquema de Relación

- $R = \text{Ingeniero} = (\text{Nombre}, \text{especialidad}, \text{antigüedad})$

Atributo (R)

Tuplas
 $r(R)$

Nombre	Especialidad	Antigüedad
José	Informático	3
Juan	Industrial	4

grado=3 cardinalidad=2

Restricciones inherentes

- Todo son **relaciones**
- Una relación **no** puede tener **tuplas duplicadas**: clave primaria
- Cada atributo únicamente puede tomar **un valor**
- Un atributo que forme parte de la clave primaria **no** puede tomar **valores nulos**

Restricciones semánticas I/II

- **Clave primaria:** PRIMARY KEY
- **Unicidad:** UNIQUE (clave candidata)
- **Obligatoriedad:** NOT NULL (atributos no vacíos)
- **Verificación:** CHECK (dominio de un atributo)

Restricciones semánticas II/II

□ Clave ajena: FOREIGN KEY

- Una clave ajena de r_1 que hace referencia a r_2 : Sean $r_1(R_1)$ y $r_2(R_2)$ relaciones con claves primarias K_1 y K_2 . Un subconjunto $A \subseteq R_1$ es **clave ajena o externa** con referencia a K_2 de r_2 si es necesario que para cada t_1 en r_1 exista una tupla t_2 en r_2 tal que $t_2[K_2] = t_1[A]$.
- La relación r_1 se denomina la **relación referenciante** de la dependencia de clave externa y r_2 se denomina **relación referenciada** de la clave ajena.

□ Integridad referencial: CASCADE, RESTRICT, SET NULL, SET DEFAULT

M. Entidad-Relación/Relacional

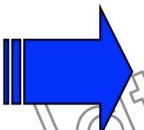
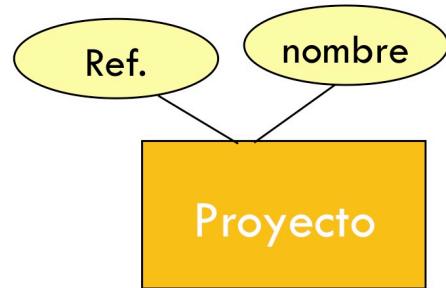
Entidad-Relación	Relacional	
Entidad	Relación	Tabla
Atributo	Atributo	Columna
Relación	Relación	Tabla
Objeto	Tupla	Fila

Conversión

- Cada elemento del modelo Entidad/Relación se convierte al modelo relacional, resultando un conjunto de tablas que se implementarán directamente en el gestor
- Notación
 - Los atributos que formen parte de la clave primaria (PK) de una tabla llevarán @ delante. ¡Cada tabla solo puede tener una PK!
 - Para las claves ajenas (FK) se indicará la tabla a la que hace referencia (se entenderá que hará referencia a la PK de la tabla a la que hace referencia).
 - Su nombre estará formado por el/los atributo/s y la tabla a la que hacen referencia (no es obligatorio, se pueden llamar de cualquier forma, pero esta forma es más organizada).
 - Es posible que una clave ajena esté formada por más de un atributo
 - Cada tabla podrá no tener ninguna FK o tener más de una FK.
 - Puede que un atributo o conjunto de atributos lleven la restricción UNIQUE.

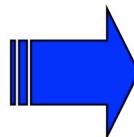
Conversión

□ Entidad



Proyecto=(@Ref., nombre)

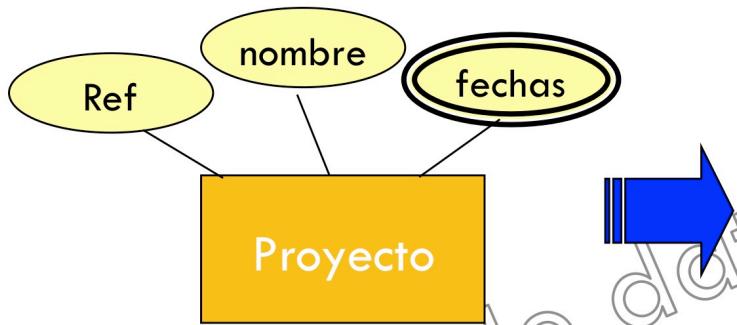
□ Atributo compuesto



Proyecto=(@Ref., dia, mes, año)

Conversión

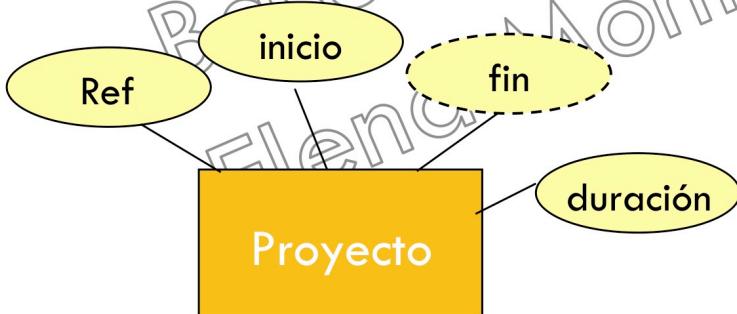
□ Atributo multivalorado



Proyecto=(@Ref., nombre)

Fechas=(@fecha, @RefProyecto (FK a
Proyecto))

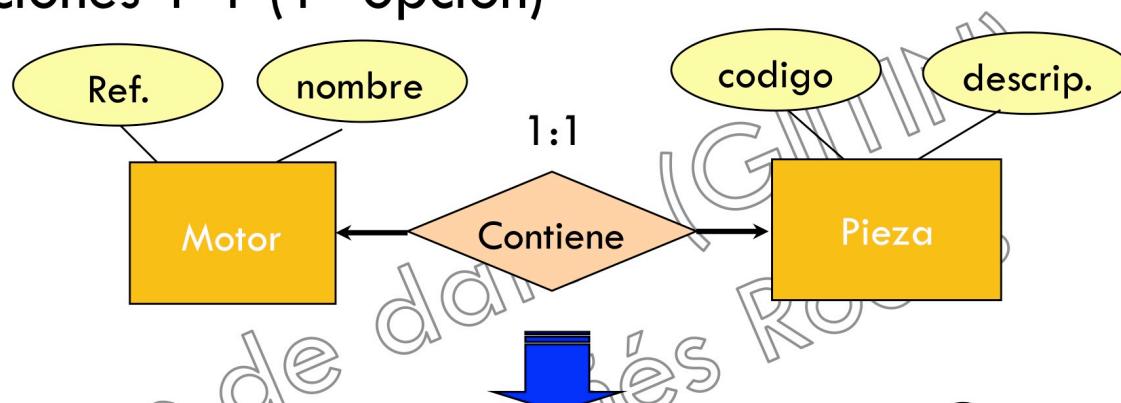
□ Atributo derivado



Proyecto=(@Ref, inicio, duración)

Conversión

□ Relaciones 1-1 (1^a opción)



Motor=(@Ref., nombre)

Pieza=@codigo, descripción)

Contiene=@RefMotor (FK a Motor), codigoPieza (Fk a Pieza) (Unique))

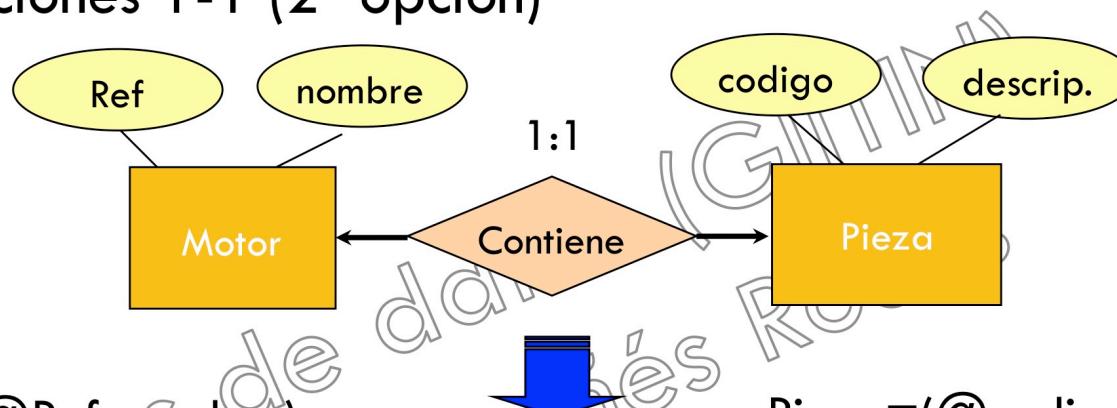
ó

Pieza=@codigo, descripción)

MotorContiene=@Ref., nombre, codigoPieza (FK a Pieza) (Unique))

Conversión

□ Relaciones 1-1 (2º opción)



Motor=(@Ref, nombre)

Pieza=@(codigo, descripción)

Contiene=(RefMotor (FK a Motor) (Unique), @codigoPieza (FK a Pieza))

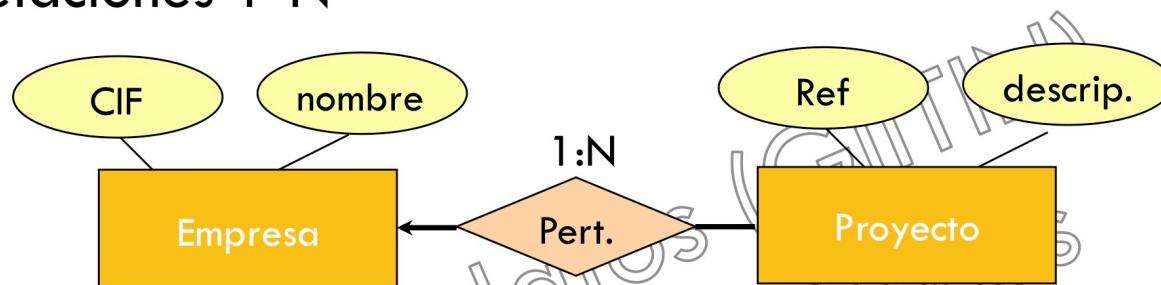
Ó

Motor=@(Ref, nombre)

PiezaContiene=@(codigo, descripción, RefMotor(FK a Motor) (Unique))

Conversión

□ Relaciones 1-N



Empresa=(@CIF, nombre)

Proyecto=@(Ref, descripción)

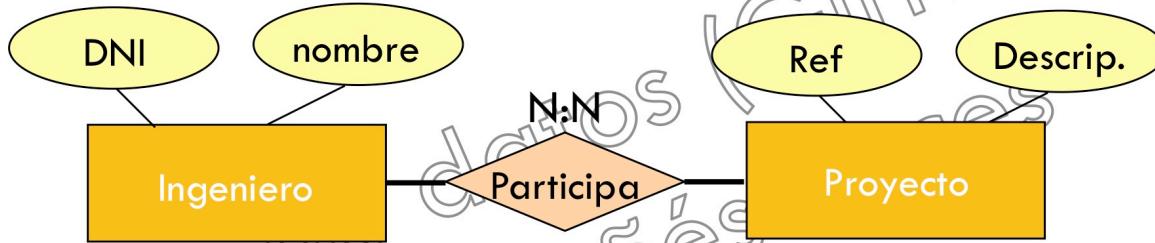
Pertenece=(CIF Empresa (FK a Empresa), @RefProyecto (FK a Proyecto))

Ó

Empresa=(@CIF, nombre) PertProy=@(Ref, descripción, CIF Empresa (FK a Empresa))

Conversión

□ Relaciones N-N



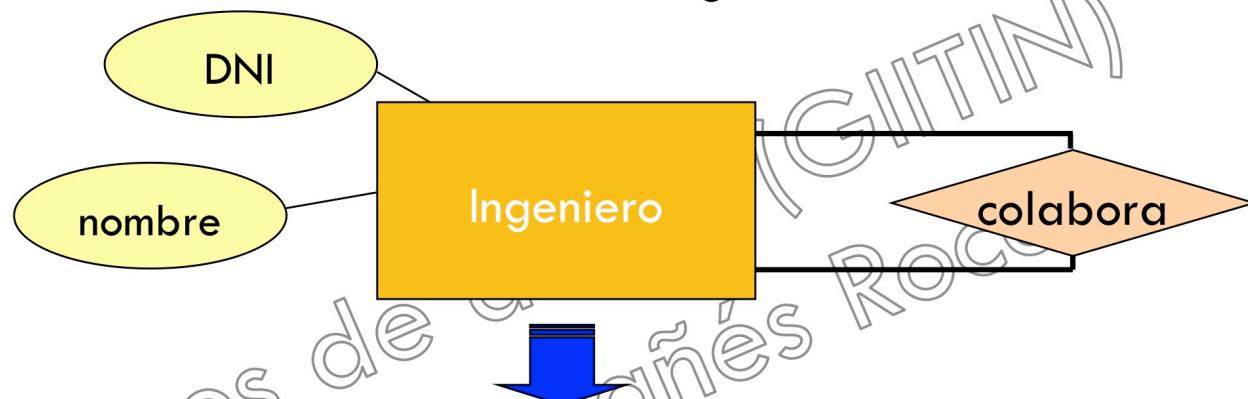
Ingeniero=(@DNI, nombre)

Proyecto=(@Ref, descripción)

Participa=(@DNI|Ingeniero (FK a Ingeniero), @Ref|Proyecto (FK a Proyecto))

Conversión

- Relaciones de entidades consigo mismas

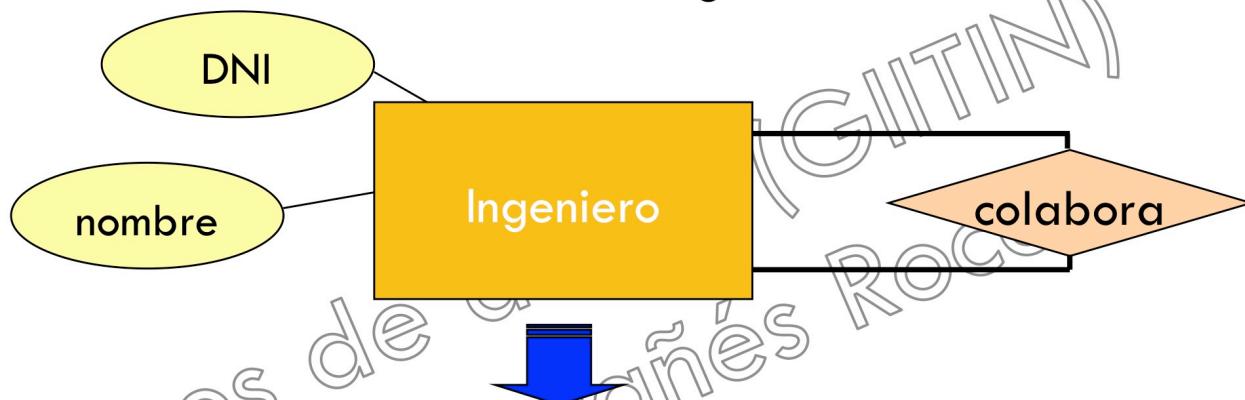


- Si la cardinalidad de colabora es N:N, entonces

Participa=(@DNIIngCoord (FK a Ingeniero), @DNIIngColab (FK a Ingeniero))

Conversión

□ Relaciones de entidades consigo mismas



- Si la cardinalidad de colabora es 1:N, entonces:

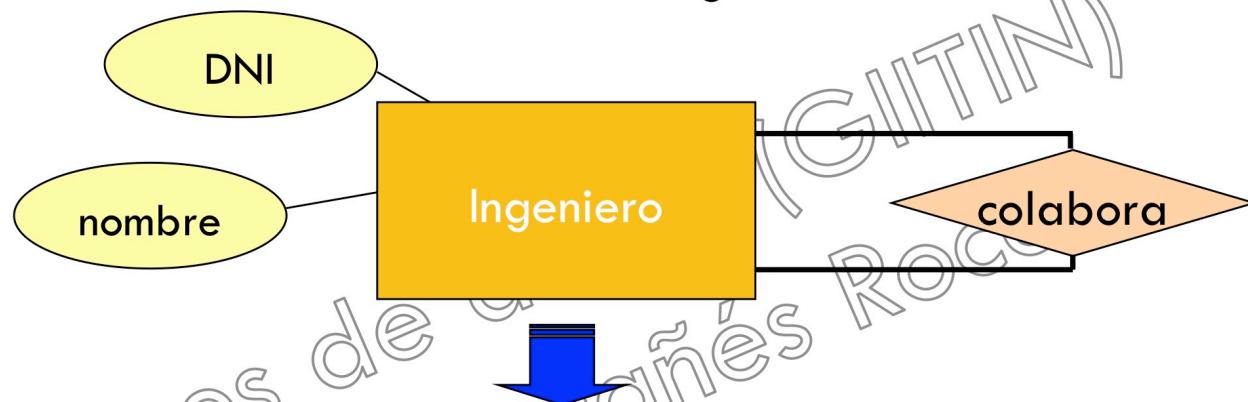
Participa=(DNI||ngCoord (FK a Ingeniero), @DNI||ngColab (FK a Ingeniero))

o se elimina la tabla participa y se modifica la tabla Ingeniero:

Ingeniero=(@DNI, nombre, DNI||ngCoord (FK a Ingeniero))

Conversión

- Relaciones de entidades consigo mismas



- Si la cardinalidad de colabora es 1:1, entonces:

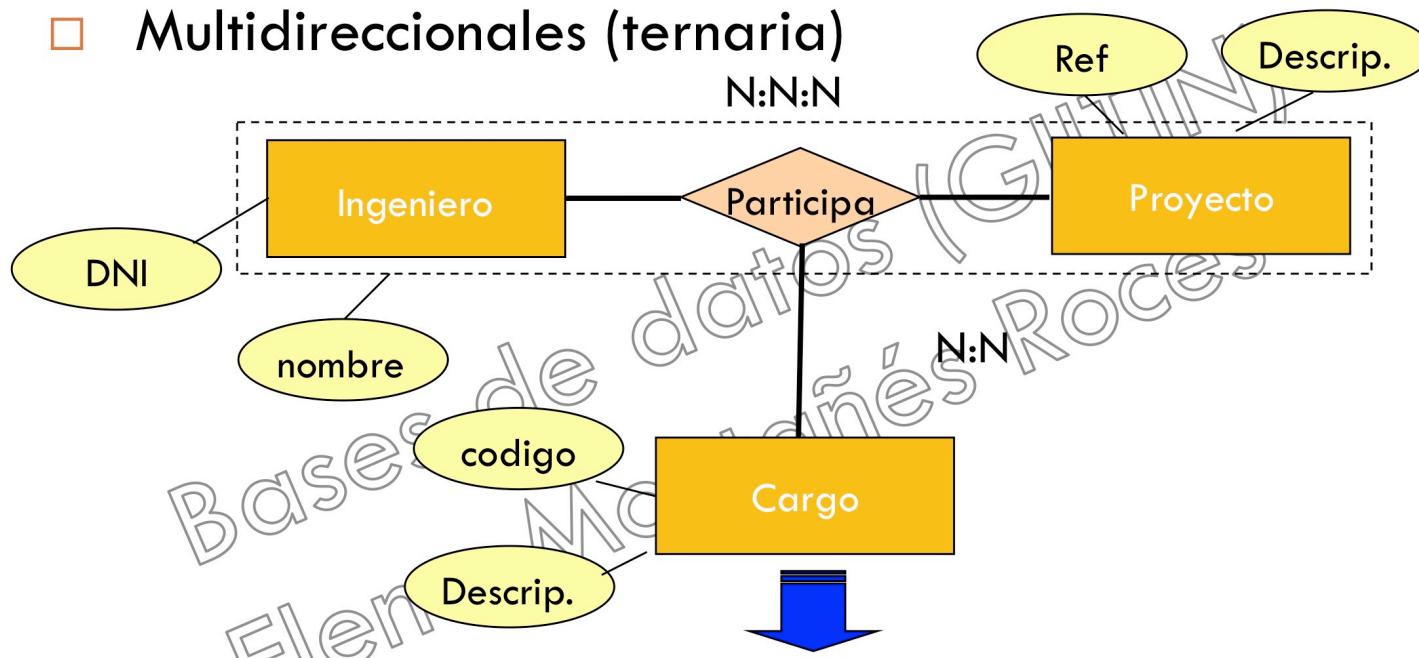
Participa=(DNI|*IngCoord* (FK a Ingeniero)(UNIQUE), @DNI|*IngColab* (FK a Ingeniero))

o se elimina la tabla participa y se modifica la tabla Ingeniero:

Ingeniero=(@DNI, nombre, DNI|*IngCoord* (FK a Ingeniero)(UNIQUE))

Conversión

□ Multidireccionales (ternaria)



Ingeniero=(@DNI, nombre) **Proyecto**=(@Ref, descripción)
Cargo=(@código, descripción)

Participa=(@DNIIngeniero (FK a Ingeniero), @RefProyecto (FK a Proyecto),
 @codigoCargo (FK a Cargo))

Conversión (sobre las multidireccionales)(I)

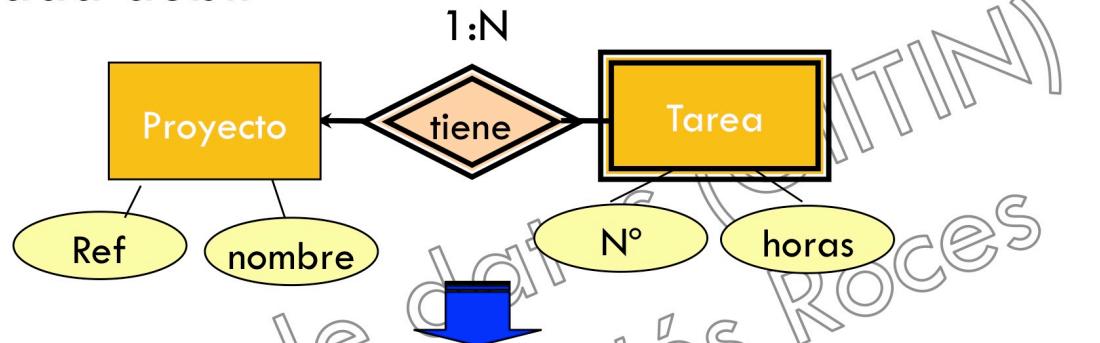
- Si la cardinalidad de la relación multidireccional (ternaria) es 1:N:N, N:1:N o N:N:1, entonces las FKs no cambian, pero se elimina de la PK de la relación, el atributo o atributos que corresponden a la PK de la parte 1, es decir:
 - Si la cardinalidad de participa es 1:N:N, entonces la PK de participa es (RefProyecto, codigoCargo)
 $\text{Participa} = (\text{DNIIngeniero (FK a Ingeniero)}, \text{@RefProyecto (FK a Proyecto)}, \text{@codigoCargo (FK a Cargo)})$
 - Si la cardinalidad de participa es N:1:N, entonces la PK de participa es (DNIIngeniero, codigoCargo)
 $\text{Participa} = (@\text{DNIIngeniero (FK a Ingeniero)}, \text{RefProyecto (FK a Proyecto)}, \text{@codigoCargo (FK a Cargo)})$
 - Si la cardinalidad de participa es N:N:1, entonces la PK de participa es (DNIIngeniero, RefProyecto)
 $\text{Participa} = (@\text{DNIIngeniero (FK a Ingeniero)}, \text{@RefProyecto (FK a Proyecto)}, \text{codigoCargo (FK a Cargo)})$

Conversión (sobre las multidireccionales)(II)

- Si la cardinalidad de la relación multidireccional (ternaria) es 1:1:N, 1:N:1 o N:1:1, entonces las FKs no cambian, y habría dos opciones para la PK:
 - Se elimina de la PK de la relación, el atributo o atributos que corresponden a la PK de una de las dos parte 1 existente. Además, la otra parte 1 existente que no forma parte de la PK junto con la parte N serían UNIQUE.
 - Se elimina de la PK de la relación, el atributo o atributos que corresponden a la PK de la otra parte 1 existente. Además, la otra parte 1 existente que no forma parte de la PK junto con la parte N serían UNIQUE.
 - Ejemplo: Si la cardinalidad de participa es 1:N:1, entonces:
 - la PK de participa puede ser (RefProyecto, codigoCargo) y entonces (DNIIngeniero, codigoCargo) sería UNIQUE
 $\text{Participa} = ((\text{DNIIngeniero } (\text{FK a Ingeniero}), @\text{RefProyecto } (\text{FK a Proyecto}))(\text{UNIQUE}),$
 $\quad @\text{codigoCargo } (\text{FK a Cargo}))$
 - la PK de participa puede ser (DNIIngeniero, RefProyecto) y entonces (RefProyecto, codigoCargo) sería UNIQUE
 $\text{Participa} = (@\text{DNIIngeniero } (\text{FK a Ingeniero}), (@\text{RefProyecto } (\text{FK a Proyecto}),$
 $\quad \text{codigoCargo } (\text{FK a Cargo}))(\text{UNIQUE}))$

Conversión

□ Entidad débil



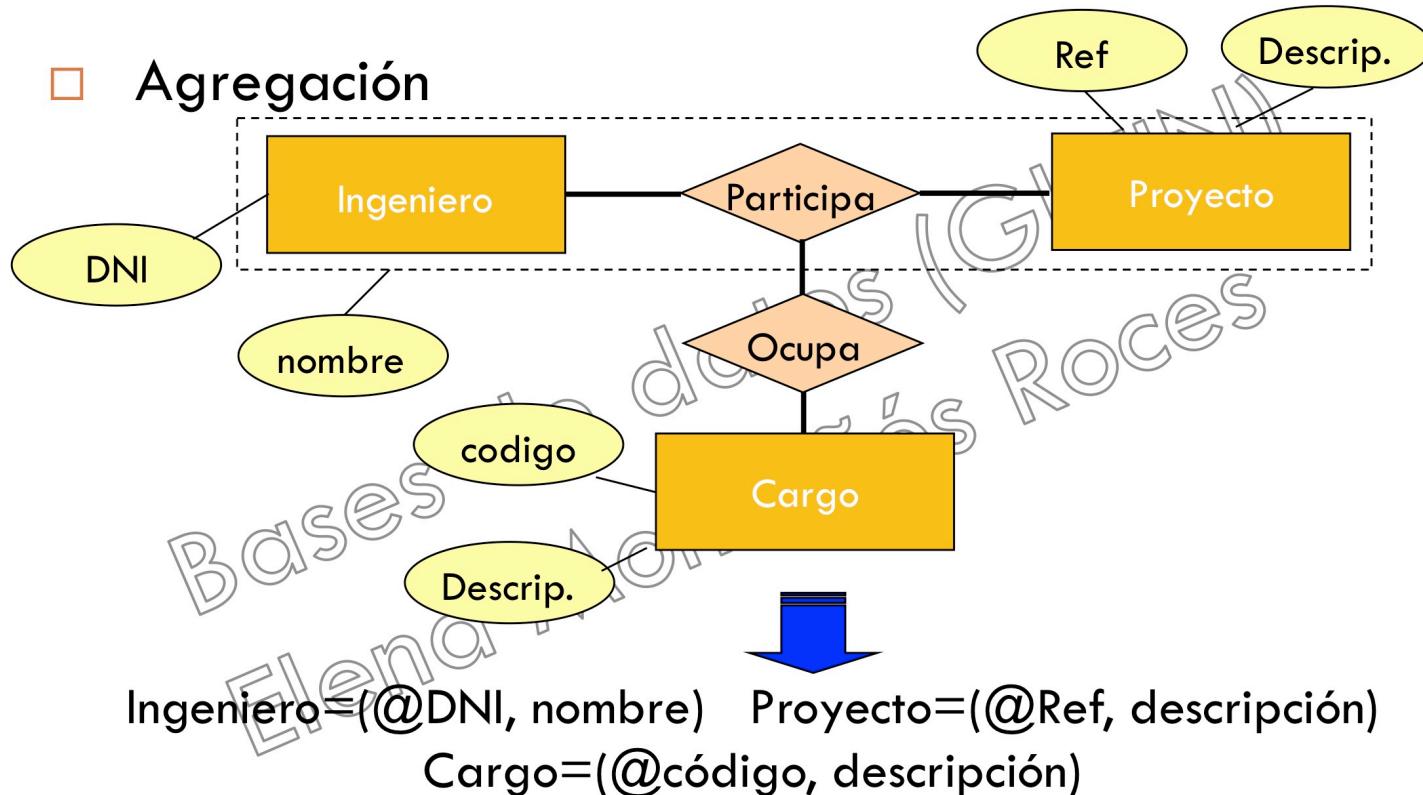
Proyecto=(@Ref, nombre)

Tarea=(@N°, horas, @RefProyecto (FK a Proyecto))

NOTA: En este caso, **NUNCA** se genera una tabla para la relación. El hecho de necesitar la P en la tabla de la entidad Tarea ya implementa la relación 1:N.

Conversión

Agregación



Participa=(@DNIIngeniero (FK a Ingeniero), @RefProyecto (FK a Proyecto))

Conversión (sobre la agregación)(I)

- La cardinalidad de la relación que agrega las entidades (**participa**) **SIEMPRE** ha de ser N:N, si no, bastaría con relacionar la tercera entidad (**cargo**):
 - con la parte N de la relación que agrega (**participa**) si fuese 1:N o N:1 o
 - con cualquiera de las partes si la relación que agrega (**participa**) fuese 1:1
- La relación que relaciona la agregación con otra entidad (**ocupa**) puede ser N:N, 1:N (o N:1) o 1:1.
 - Las FKs no cambian
 - Solo cambia la PK

Conversión (sobre la agregación)(II)

- Si la relación que relaciona la agregación con otra entidad (ocupa) es N:N, entonces:
 - la PK de ocupa está formada por la unión de las PK de participa y cargo (DNIIngenieroParticipa, RefProyectoParticipa, codigoCargo) (ver ejemplo anterior)
 $Ocupa=((@DNIIngenieroParticipa, @RefProyectoParticipa)$
 $(FK a Participa), @codigoCargo (FK a Cargo))$

Conversión (sobre la agregación)(III)

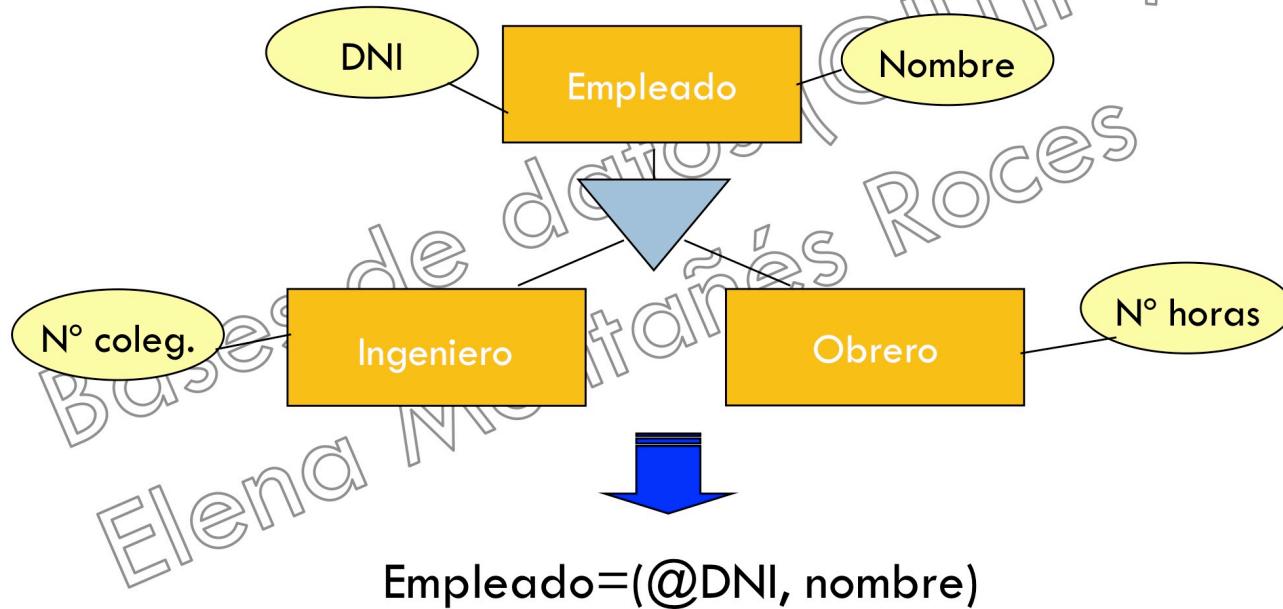
- Si la relación que relaciona la agregación con otra entidad es 1:N o N:1, entonces:
 - Si es 1:N, la PK de ocupa estaría formada solo por la PK de la parte N, es decir, solo por la PK de cargo (codigoCargo).
 $Ocupa=((DNIIngenieroParticipa, RefProyectoParticipa) \text{ (FK a Participa)}, @codigoCargo \text{ (FK a Cargo)})$
 - Si es N:1, la PK de ocupa estaría formada solo por la PK de la parte N, es decir, solo por la PK de participa (DNIIngenieroParticipa, RefProyectoParticipa)
 $Ocupa=((@DNIIngenieroParticipa, @RefProyectoParticipa) \text{ (FK a Participa)}, codigoCargo \text{ (FK a Cargo)})$

Conversión (sobre la agregación)(IV)

- Si la relación que relaciona la agregación con otra entidad es 1:1, hay dos opciones:
 - Que la PK de ocupa podría estar formada solo por la PK de cargo (codigoCargo) y entonces la PK de participa (DNIIngenieroParticipa, RefProyectoParticipa) sería UNIQUE
 $Ocupa=((DNIIngenieroParticipa, RefProyectoParticipa) \text{ (FK a Participa)} \text{ (UNIQUE)}, @codigoCargo \text{ (FK a Cargo)})$
 - Que la PK de ocupa podría estar formada solo por la PK de participa (DNIIngenieroParticipa, RefProyectoParticipa) y entonces la PK de cargo sería UNIQUE
 $Ocupa=((@DNIIngenieroParticipa, @RefProyectoParticipa) \text{ (FK a Participa)}, codigoCargo \text{ (FK a Cargo)} \text{ (UNIQUE)})$

Conversión

□ Especialización/Generalización general

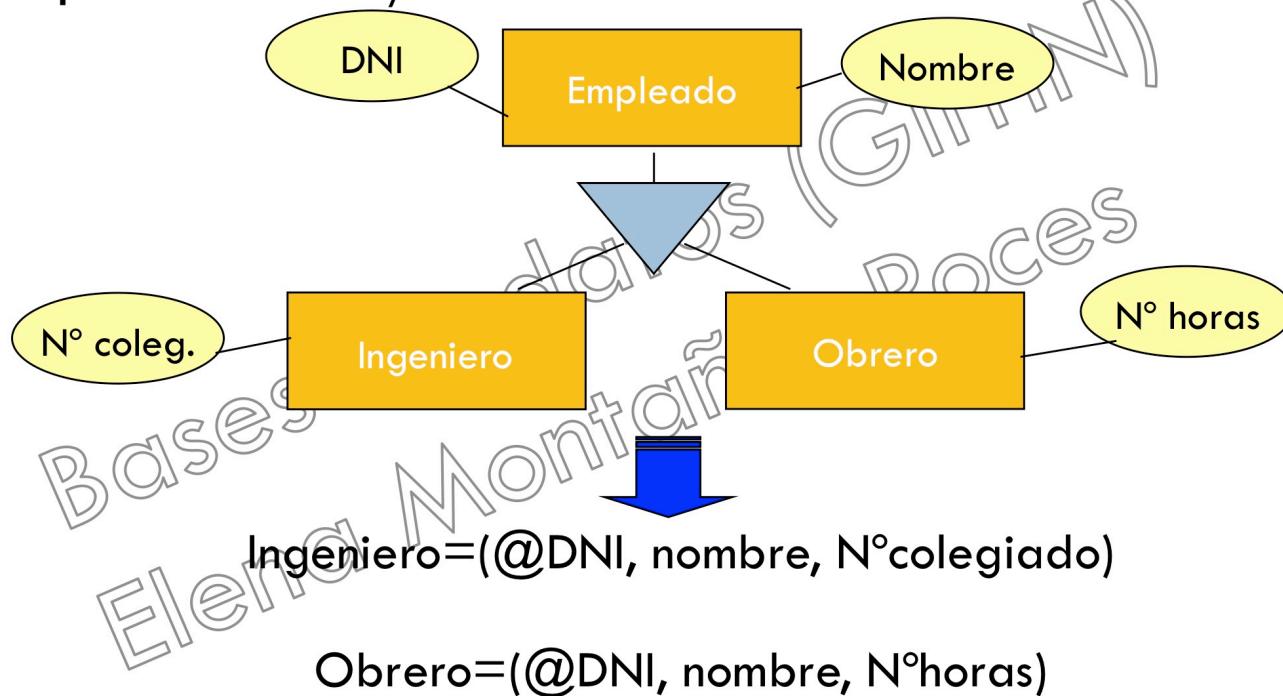


Ingeniero=(@DNI Empleado (FK a Empleado), Nºcolegiado)

Obrero=(@DNI Empleado (FK a Empleado), Nºhoras)

Conversión

□ Especialización/Generalización total



- Si **disjunta**, entonces **no** se repite información
- Si **solapada**, entonces **sí** se repite información