

# Unidad 3

## Esquema Relacional

# Del diseño conceptual al diseño lógico

## ► **Restricciones** del modelo relacional:

1. En una relación no puede haber dos tuplas iguales
2. El orden de las tuplas y de los atributos no es relevante

El orden de las columnas importa cuando se hace la inserción de las filas

Ambas tablas son iguales desde el punto de vista lógico

CodAsig	Nombre	NumHoras	CodCF
1	Base de datos	165	1
2	Lenguaje de marcas	120	1
3	Seguridad informática	90	2
4	Despliegue aplicaciones webs	110	1
5	Fundamentos de hardware	90	2
6	Acceso a datos	180	1

CodAsig	CodCF	NumHoras	Nombre
1	1	165	Base de datos
2	1	120	Lenguaje de marcas
3	2	90	Seguridad informática
4	1	110	Despliegue aplicaciones webs
5	2	90	Fundamentos de hardware
6	1	180	Acceso a datos

# Transformación de entidad y atributos

## Transformación de atributos

- ▶ **Cada atributo** de una entidad se transforma en **una columna** de una tabla.
- ▶ Los **atributos principales** pasan a formar parte de la **PK** de la tabla.
- ▶ Los **atributos secundarios** serán claves secundarias con restricción de **unicidad**.
- ▶ Los **atributos obligatorios** tendrán la restricción **NOT NULL** (no se indica con ninguna letra).
- ▶ Los **atributos multivaluados** se convierten en una **nueva tabla**:

- Clave primaria : concatenación de la clave primaria de la tabla principal y valor
- Clave ajena referenciando la tabla principal

ALUMNO
P·dni
U·NSS
Nombre
Ap1
N·Ap2
MN·Telefono

Alumno (P·dni, nombre, U·NSS, ap1, N·ap2)  
Telefono (P·numTelefono, PF·dni -> alumno)

# Transformación de entidad y atributos

## Transformación de atributos

- ▶ Los **atributos derivados** son atributos con un **cálculo asociado** a cada ocurrencia
- ▶ Los **atributos compuestos** se trasladan a la tabla **añadiendo una columna** por cada atributo que lo compone.
- ▶ Los **atributos codificables** (con dominio finito y conocido y susceptibles de extraerse a una entidad específica) se convierten en una **nueva tabla**:
  - ▶ El nombre de la tabla es el nombre del atributo
  - ▶ La PK es una secuencia
  - ▶ Añaden campo descripción, con los valores posibles del atributo.

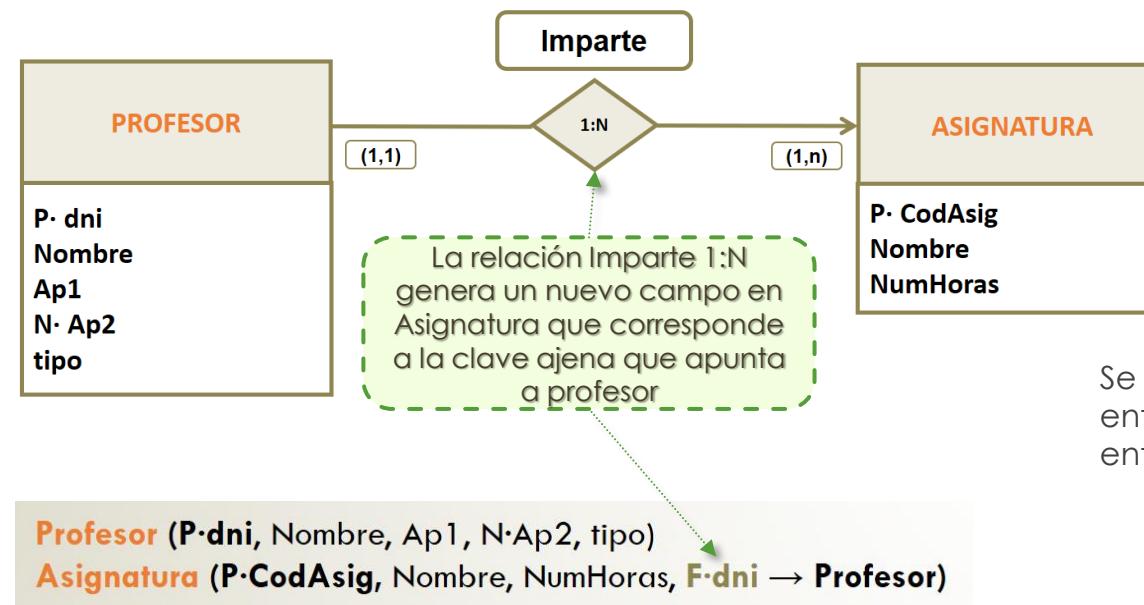
# Transformación de entidad y atributos

- ▶ **Notación:** Nombre de la relación y a continuación los nombres de atributos, entre paréntesis y separados por comas)
  - ▶ Ejemplo:

**Profesor** (**P**·dni, **U**·NSS, nombre, primerApell, **N**·SegApell, **F**·codCF -> CicloFormativo, **F**·codDpto -> Departamento)

# Transformación de las relaciones

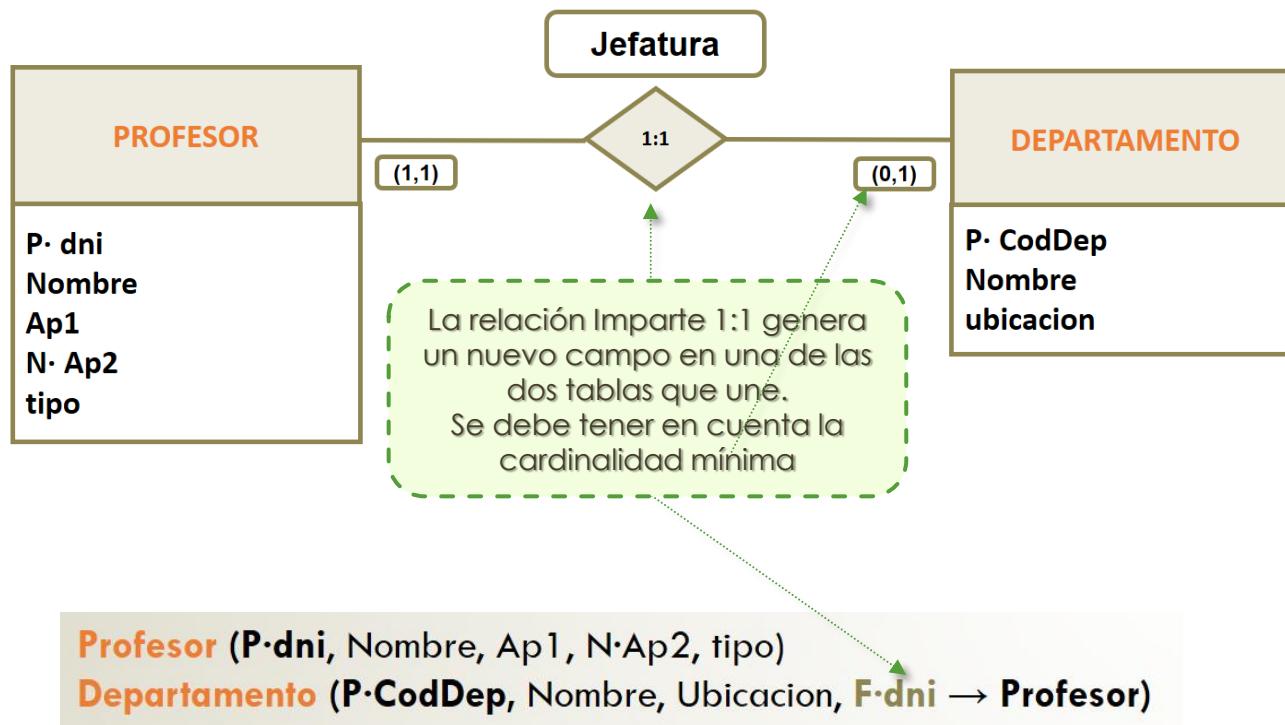
## ▶ Transformación de la relación 1:N



Se propaga el campo clave de la entidad con cardinalidad menor a la entidad con cardinalidad más alta

# Transformación de las relaciones

## ▶ Transformación de la relación 1:1

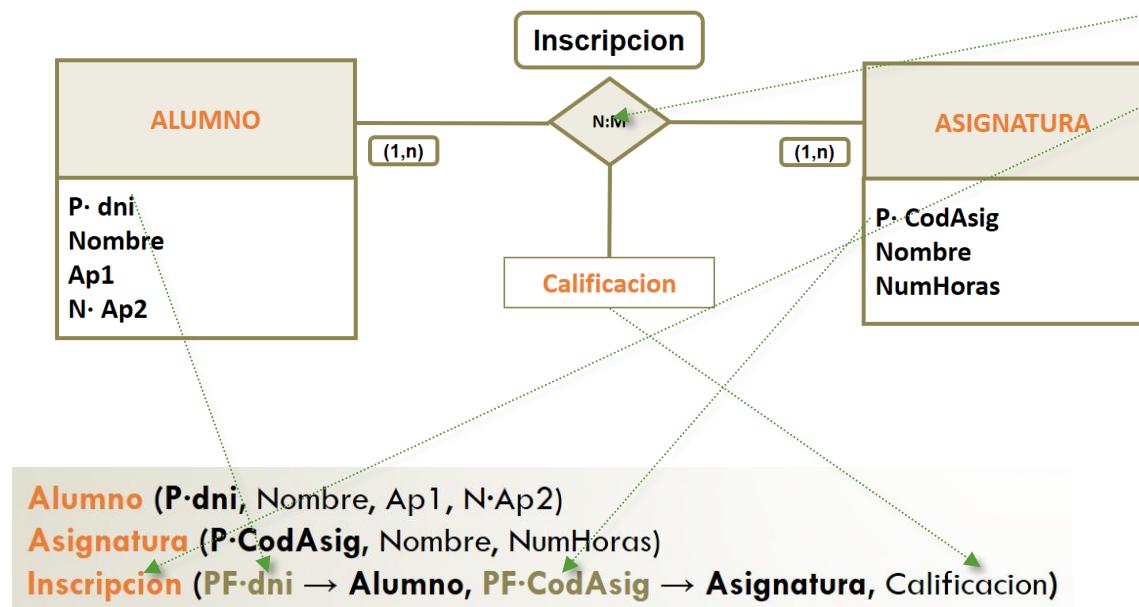


En las relaciones 1:1 se podría propagar la PK a cualquiera de las tablas (en cualquier sentido). Pero en este caso interesa la combinación que genera menos FK con valor a null.

Un departamento siempre va a tener un profesor jefe, por lo tanto, la FK en Departamento siempre está informada. Sin embargo, un profesor puede no ser jefe de ningún departamento, de hecho, la mayoría no lo es, si se colocase la FK en 'Profesor' hacia 'Departamento', en muchas tuplas la FK tendría valor null.

# Transformación de las relaciones

## ► Transformación de la relación N:M



La relación Inscripción  
N:M genera una nueva  
tabla.

Se crea una nueva tabla con PK formada  
por las PK de las tablas que se relacionan