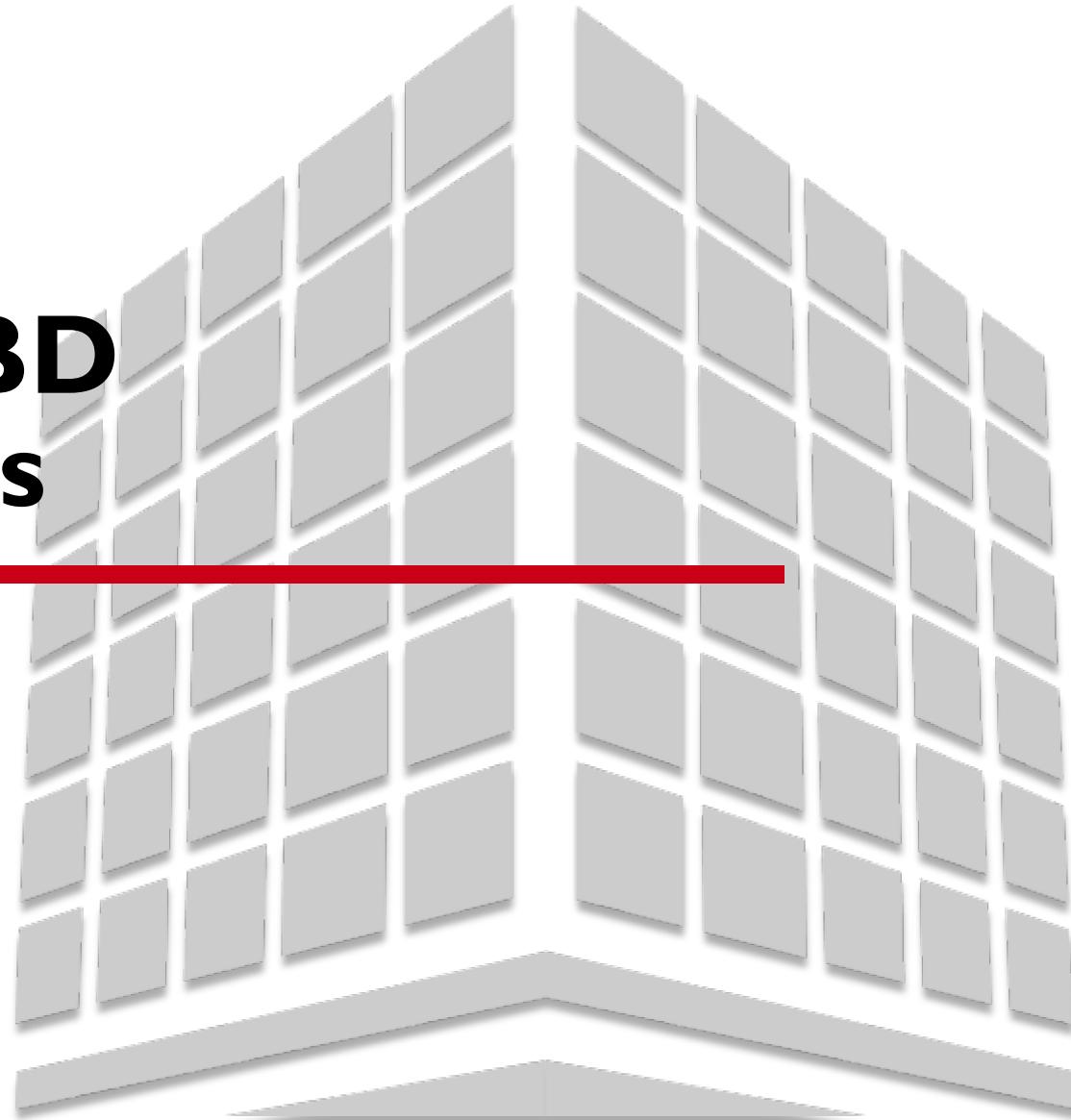


Diseño de BD Relacionales





Diseño de Bases de Datos Relacionales

Contenido

Tema 7: Diseño de Bases Datos Relacionales

7.1 Metodología de Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar

 7.2.1 Reglas de Transformación

7.3 Diseño Lógico Específico y Físico

7.4 Teoría de Normalización



Diseño de Bases de Datos Relacionales

Bibliografía Básica

- **Tecnología y Diseño de Bases de Datos.**
 - M. Piattini, E. Marcos, C. Calero y B. Vela, Ed.: RAMA. 2006
⇒ PARTE IV – Capítulos 13-17



Diseño de Bases de Datos Relacionales

Bibliografía Complementaria

- **Fundamentals of Database Systems (7^a edición).**
 - Elmasri, R. y Navathe, S.B. Addison-Wesley, 2015
- **Database System Concepts (7^a edición).**
 - A. Silberschatz, H. F. Korth y S. Sudarshan, Mc Graw-Hill, 2010
- **Database Systems: A Practical Approach to Design, Implementation and Management** 4
 - T. Connolly y C. Begg, Global Edition, 2014.
- **An Introduction to Database Systems (8^a edición).**
 - Date, C. J., Addison-Wesley, 2004.
- **Databases Illuminated (3^a edición).**
 - Ricardo, Jones & Bartlett Learning, 2015.
- **Information Modeling and Relational Databases (2^a edición)**
 - T. Halpin y T. Morgan. Morgan Kaufmann, 2010.
- **Database Design and Relational Theory: Normal Forms and All That Jazz.**
 - C.J. Date, O'Reilly Media, Inc, 2012.



Diseño de Bases de Datos Relacionales

7.1 Metodología de Diseño de Bases de Datos Relacionales

Para plasmar nuestro universo del discurso utilizando el **Modelo Relacional** se puede proceder de dos modos distintos:

- a) Obtener **directamente** las relaciones, atributos y restricciones a partir del análisis del mundo real: **esquema relacional**.
- b) Utilizar una aproximación **metodológica** a partir de un **esquema conceptual**.



Diseño de Bases de Datos Relacionales

7.1 Metodología de Diseño de Bases de Datos Relacionales

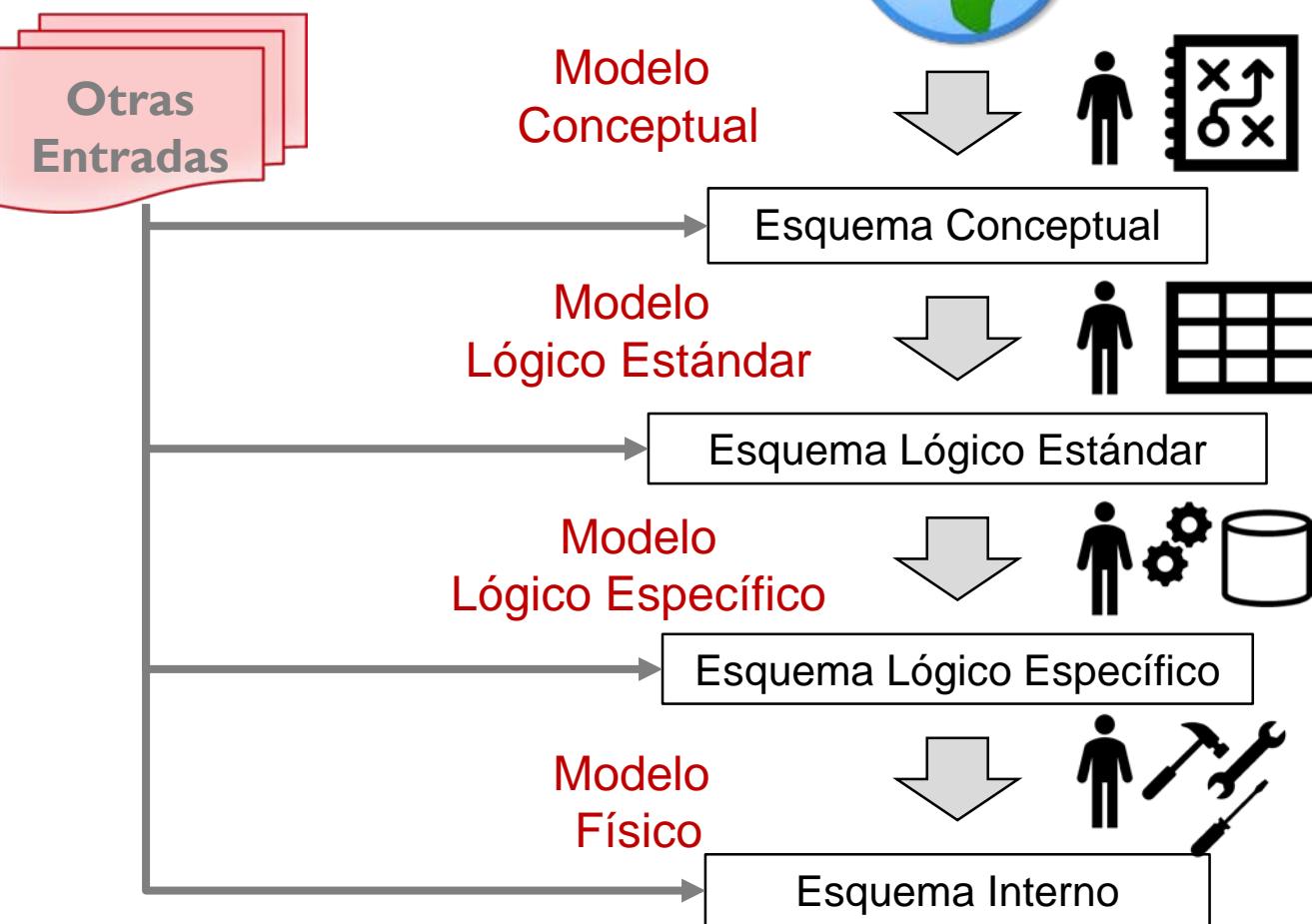
Para plasmar nuestro universo del discurso utilizando el **Modelo Relacional** se puede proceder de dos modos distintos:

- a) Obtener **directamente** las relaciones, atributos y restricciones a partir del análisis del mundo real: **esquema relacional**.
- b) Utilizar una aproximación **metodológica** a partir de un **esquema conceptual**.

En cualquiera de estos dos modos **pueden producirse fallos**. Por este motivo se debe analizar cuidadosamente el esquema final obtenido, utilizando la **teoría de la normalización**.

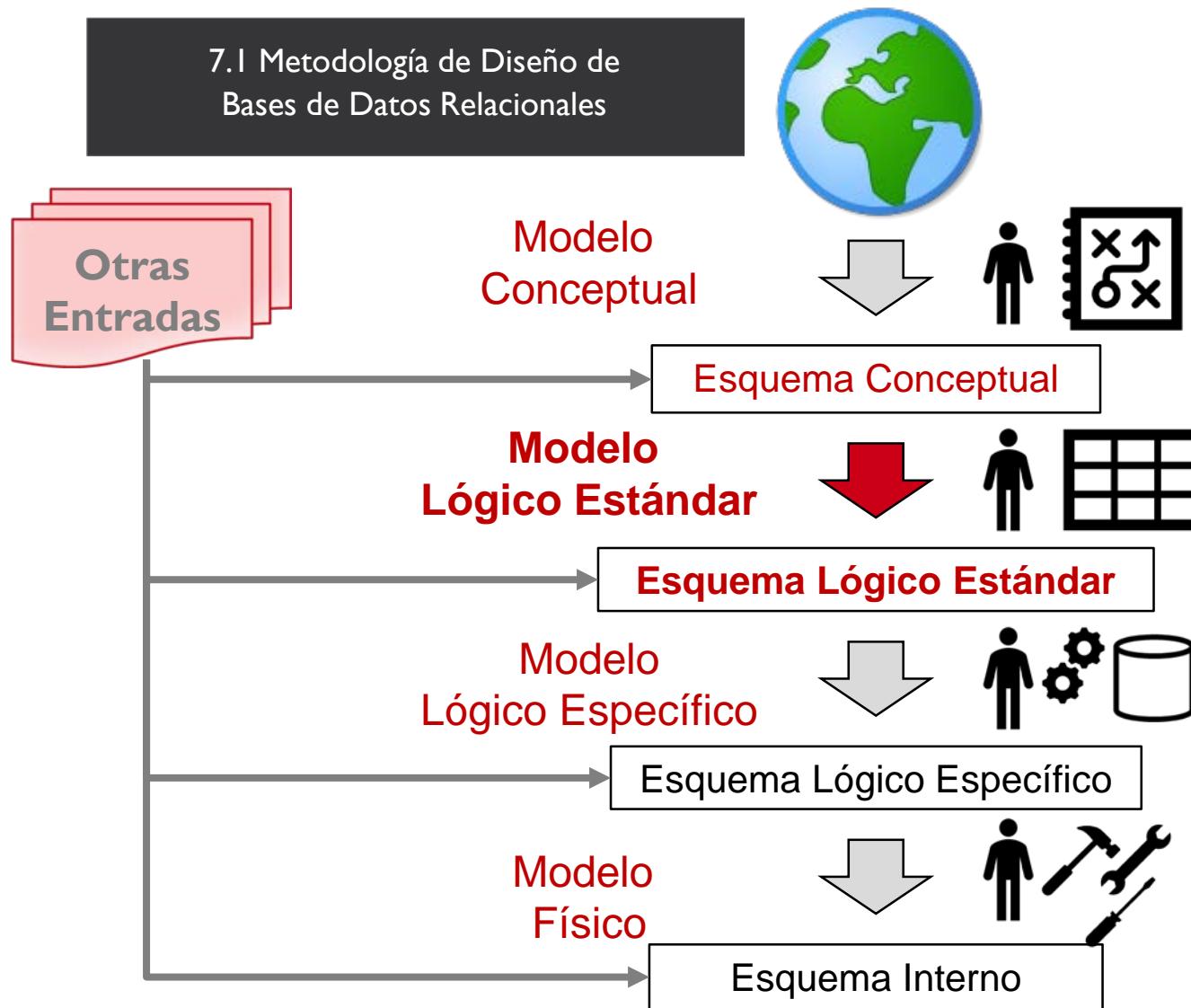
Diseño de Bases de Datos Relacionales

7.1 Metodología de Diseño de Bases de Datos Relacionales



Diseño de Bases de Datos Relacionales

7.1 Metodología de Diseño de Bases de Datos Relacionales





Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Transformación de un Esquema E/R a un Esquema Relacional

Objetivo:

Convertir el esquema conceptual E/R en un esquema lógico relacional.

12

- Aplicando adecuadamente las **reglas de transformación** se obtiene un **esquema lógico relacional normalizado**.

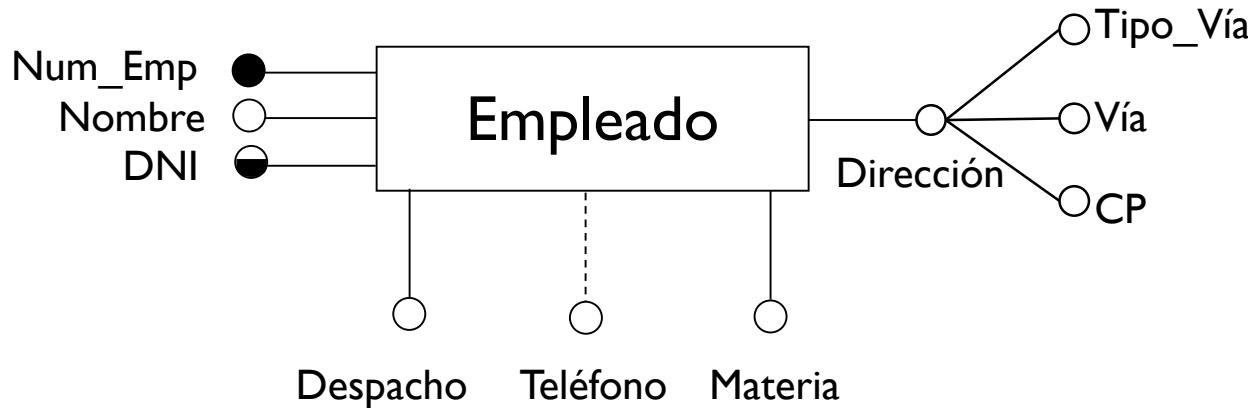
- A partir del esquema lógico específico se obtendrá el **esquema interno**. Este esquema, por motivos de **rendimiento y optimización, puede ser desnormalizado**.

7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Recordatorio M E/R

Modelo Conceptual: E/R Extendido

Entidades y Atributos



Interrelaciones

I:N

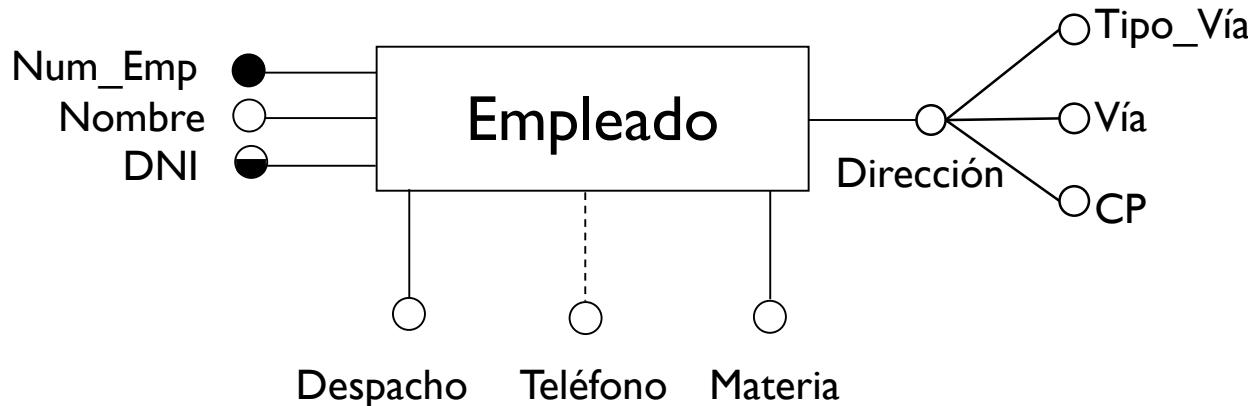


7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Recordatorio M E/R

Modelo Conceptual: E/R Extendido

Entidades y Atributos



Interrelaciones

N:M





Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar
7.2.1 Reglas de Transformación

Recordatorio MR

Grafo Relacional

- Claves primarias: subrayado
- Claves alternativa: **negrita** (UNIQUE)
- Claves ajenas: *cursiva* o subrayado discontinuo
- Atributos que admiten valores nulos: asterisco (*)
- Atributos que no admiten valores nulos: NOT NULL (NN)
- Opciones de borrado y modificación: restringido (R), en cascada (C), puesta a nulos (N), puesta a valor por defecto (D)

B:R, M:C

PROFESOR (Cod_Profe, Nombre^{NN}, **DNI**^{NN}, Direccion*, Materia*, *Cod_Area**)

→ **AREA** (Cod_Area, Nombre_Area^{NN}, Descripción*)

Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

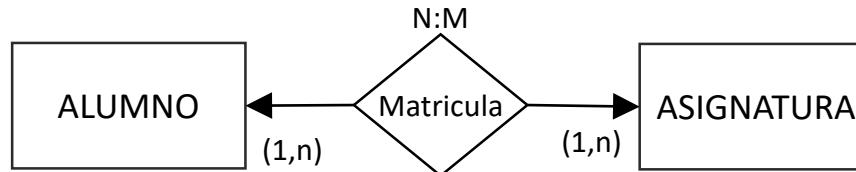
□ Reglas básicas:

• Entidad

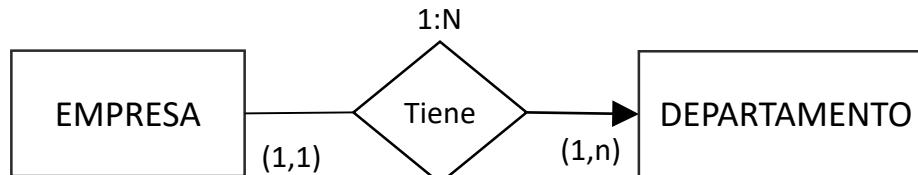
- Todo tipo de **entidad** se convierte en una **relación**, incluyendo sus atributos como atributos de la relación.

• Interrelación

- Todo tipo de **interrelación N:M** se transforma en una **relación**.



- Para todo tipo de **interrelación 1:N** se realiza lo que se denomina **propagación de clave** (regla general), o se crea una nueva **relación**.



PÉRDIDA DE SEMÁNTICA:

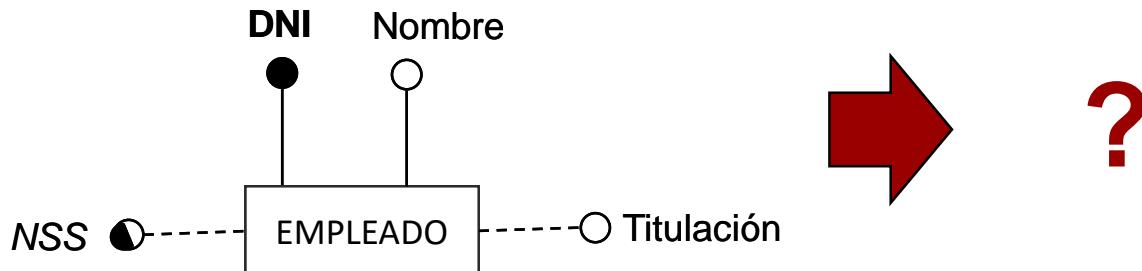
- El Modelo Relacional **no** distingue entre entidades e interrelaciones.
- En las **I:N** se puede **perder** hasta el nombre de la interrelación.

Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Entidad: Transformación de Atributos

- **Atributos de entidades ⇒ Columnas de la relación**
 - Identificador Principal ⇒ Clave Primaria
 - Identificador Alternativo ⇒ UNIQUE, ¿NOT NULL?
 - No identificadores ⇒ columnas, ¿NOT NULL?



Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Entidad: Transformación de Atributos

- **Atributos de entidades ⇒ Columnas de la relación**
 - Identificador Principal ⇒ Clave Primaria
 - Identificador Alternativo ⇒ UNIQUE, ¿NOT NULL?
 - No identificadores ⇒ columnas, ¿NOT NULL?

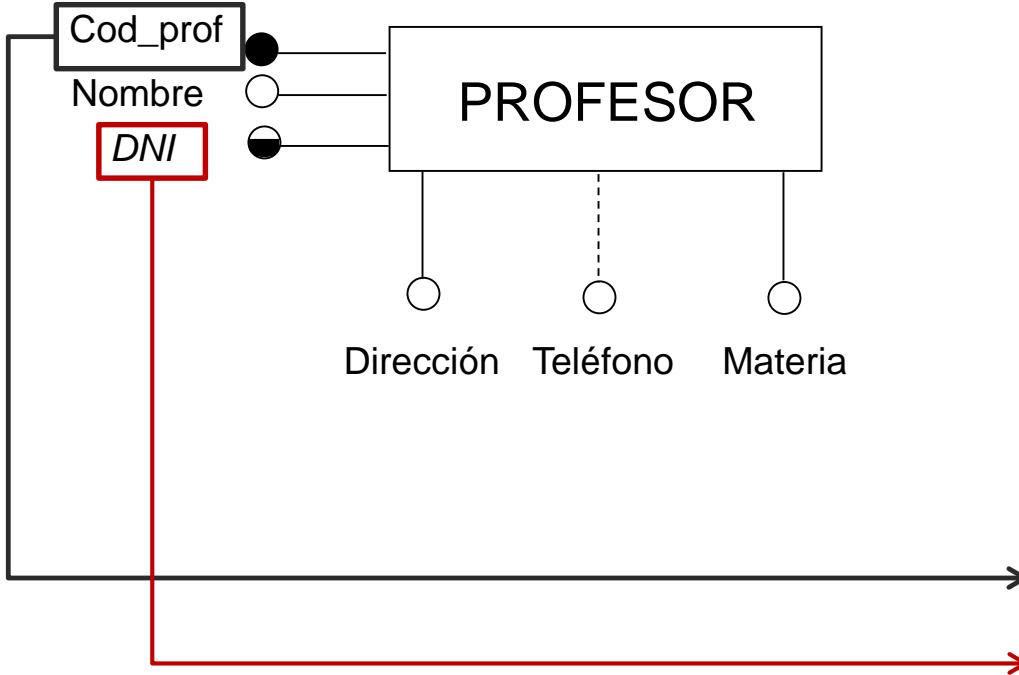


Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar

7.2.1 Reglas de Transformación

Entidad: Transformación de Atributos



Sentencia de creación de la relación PROFESOR en SQL:

```
CREATE TABLE Profesor(  
    Cod_prof Códigos,  
    Nombre Nombres NOT NULL,  
    DNI DNIS NOT NULL,  
    Dirección Lugares NOT NULL,  
    Teléfono Nos_Teléfono,  
    Materia Materias NOT NULL,  
    PRIMARY KEY (Cod_Prof),  
    UNIQUE (DNI));
```

PROFESOR (Cod_prof, Nombre^{NN}, **DNI**^{NN}, Dirección^{NN}, Teléfono*, Materia^{NN})

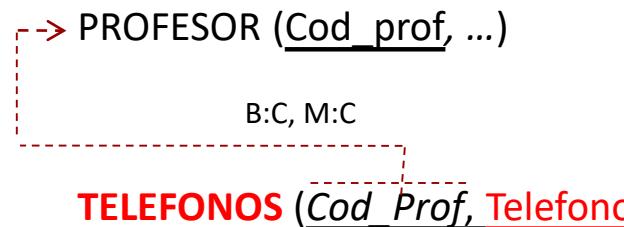
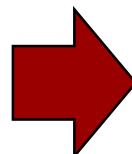
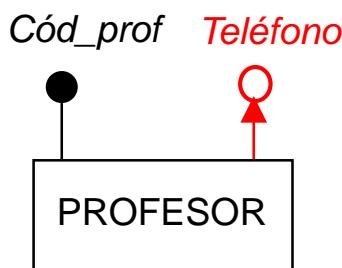
Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Entidad: Transformación de Atributos

Atributos Multivaluados

De Entidad:



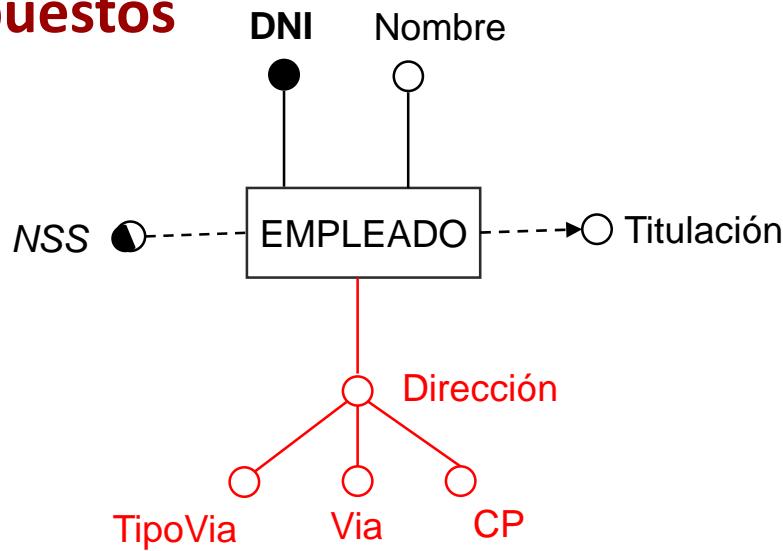
¿Qué ocurre si el teléfono es personal (no compartido)?

Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Entidad: Transformación de Atributos

Atributos Compuestos



22

→ **EMPLEADO** (DNI, Nombre^{NN}, NSS^{UNIQUE*}, TipoVia^{NN}, Via^{NN}, CP^{NN})

B:C, M:C

TITULACION (DNI_Emp, Titulacion)



Diseño de Bases de Datos Relacionales

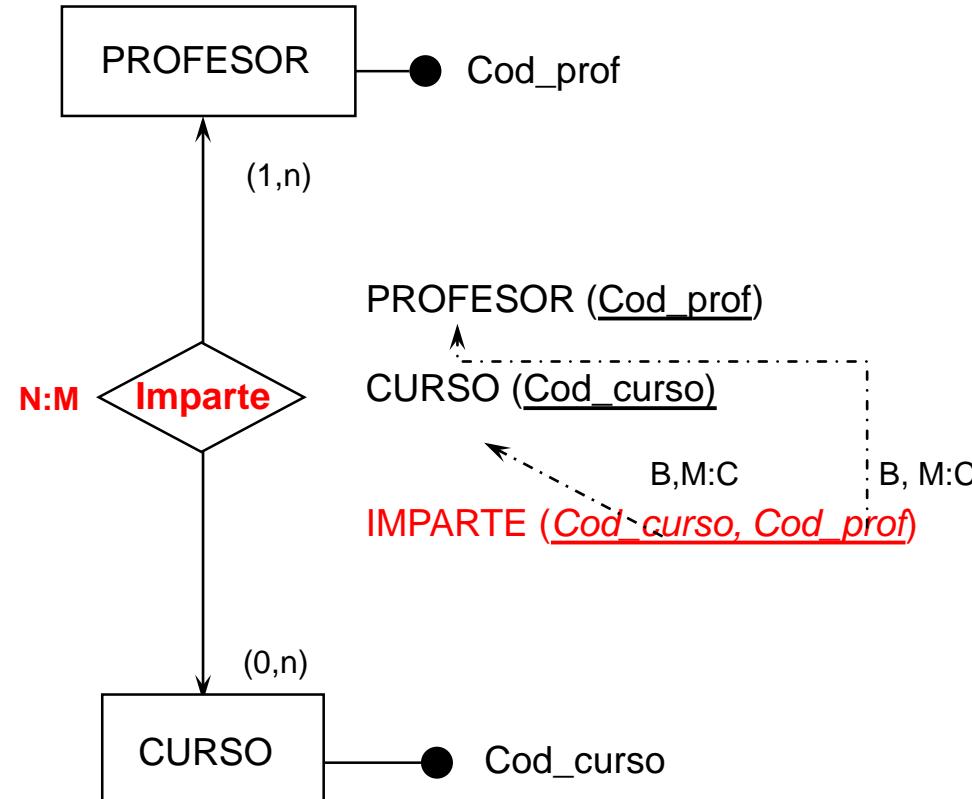
7.2 Diseño Lógico Estándar

7.2.1 Reglas de Transformación

Interrelaciones

- **Interrelaciones:**
Según tipo de correspondencia y otros aspectos semánticos
- **N:M ⇒ Nueva Relación**
 - **Clave primaria:** concatenación Identificador Principal de los tipos de entidad que asocia, que también serán claves ajenas.
 - **Opciones de borrado y modificación:** restringido o en cascada (¿Puesta a nulo, puesta a valor por defecto?)
 - **Cardinalidades mínimas:** restricciones, aserciones o disparadores (¿estándar? ¿productos?).

Diseño de Bases de Datos Relacionales



7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Interrelaciones

Sentencia de creación de la relación Imparte en SQL:

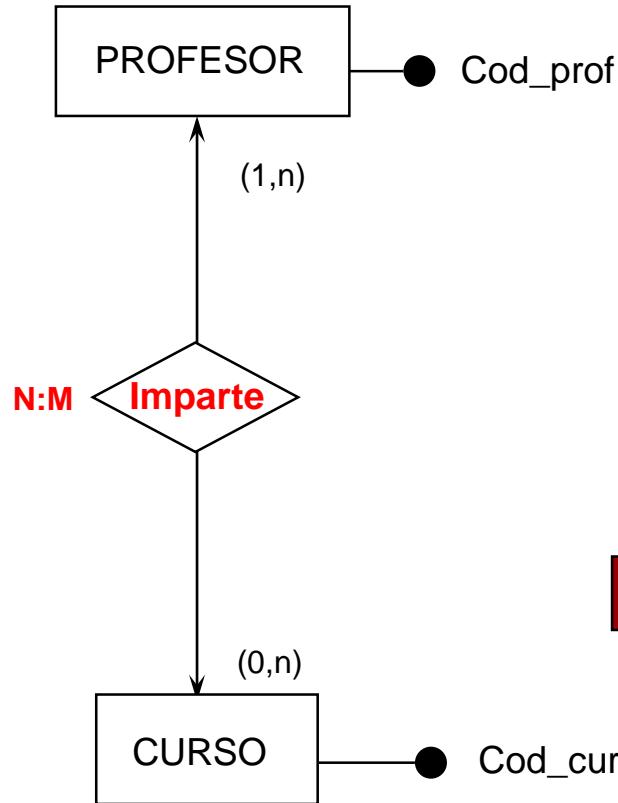
```
CREATE TABLE Imparte(
    Cod_prof Codigos_P,
    Cod_curso Codigos_C,
    PRIMARY KEY (Cod_prof, Cod_curso),
    FOREIGN KEY (Cod_prof)
        REFERENCES Profesor
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (Cod_curso)
        REFERENCES Curso
        ON DELETE CASCADE
        ON UPDATE CASCADE));
```

Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar

7.2.1 Reglas de Transformación

Interrelaciones

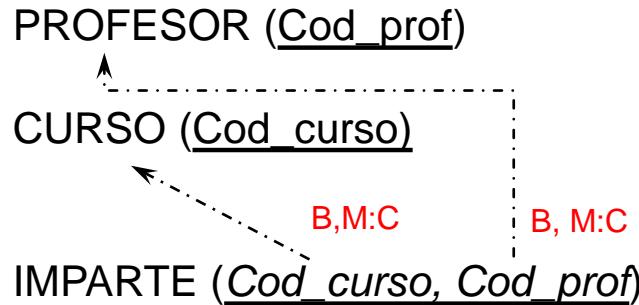


- Opciones de Borrado y Modificación: serían válidas otras opciones **pero no SET NULL** ¿Por qué?

Pertenece a la Clave Primaria: no acepta Nulos
(restricción inherente del Modelo Relacional)

- ¿Cómo controlamos la **cardinalidad mínima** de cada una de las entidades que participan en la interrelación?

Restricciones o disparadores

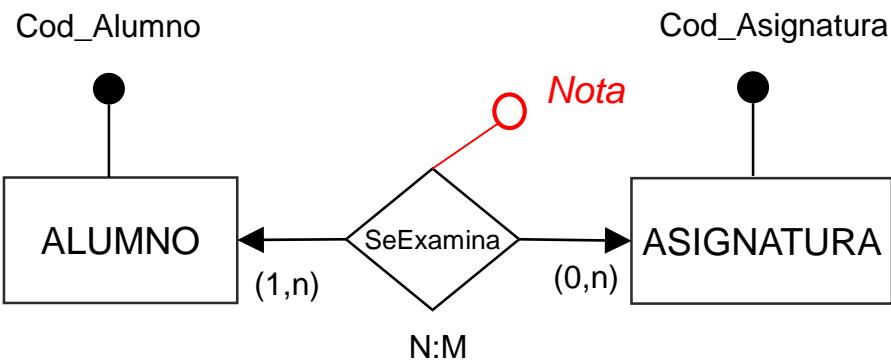


Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Atributos en Interrelaciones

26



--> ALUMNO (Cod_Alumno)

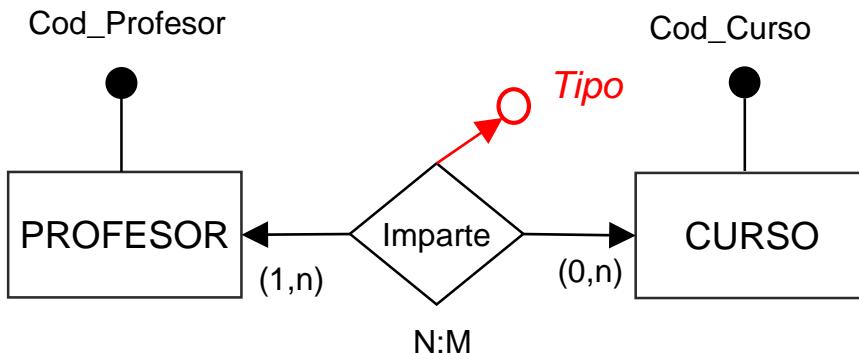
SeExamina(Cod_Asig, Cod_Alum, Nota^{NN})

--> ASIGNATURA (Cod_Asignatura)

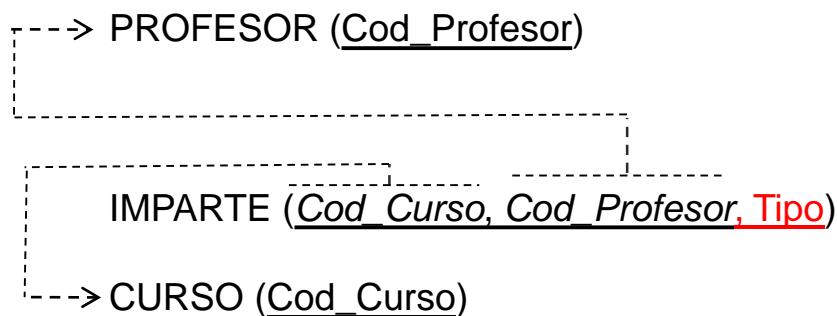
Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Atributos en Interrelaciones



Tipo de Docencia = (Práctica o Teoría)



Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar

7.2.1 Reglas de Transformación

Interrelaciones

□ **I:N** \Rightarrow 2 posibilidades:

- A. Propagar los Identificadores Principales en el sentido de la flecha (**regla básica**)
- B. Transformarlo en una relación



Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar
7.2.1 Reglas de Transformación

Interrelaciones

1:N \Rightarrow 2 posibilidades:

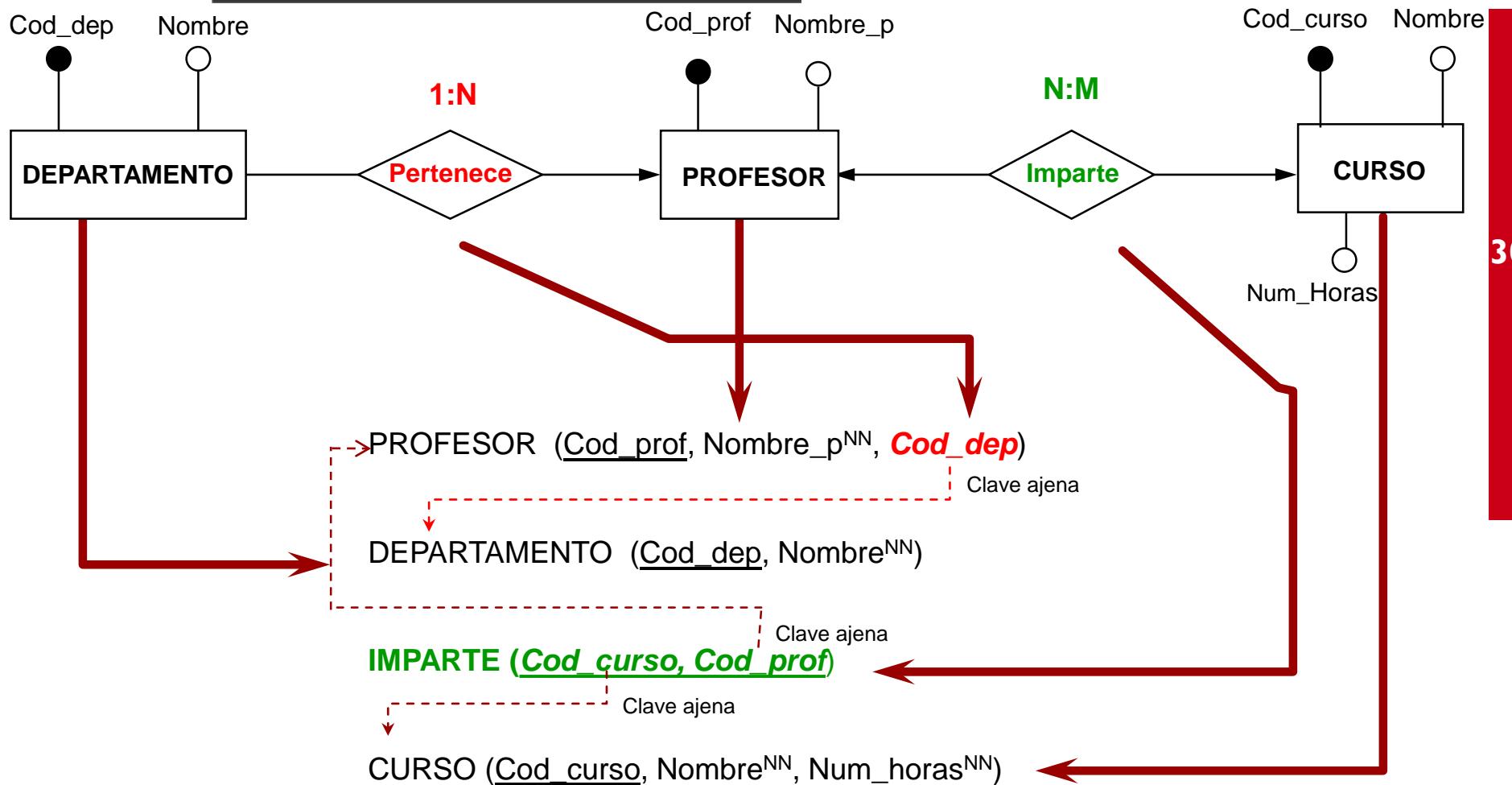
- A. Propagar los Identificadores Principales en el sentido de la flecha (**regla básica**)
- B. Transformarlo en una relación

Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar

7.2.1 Reglas de Transformación

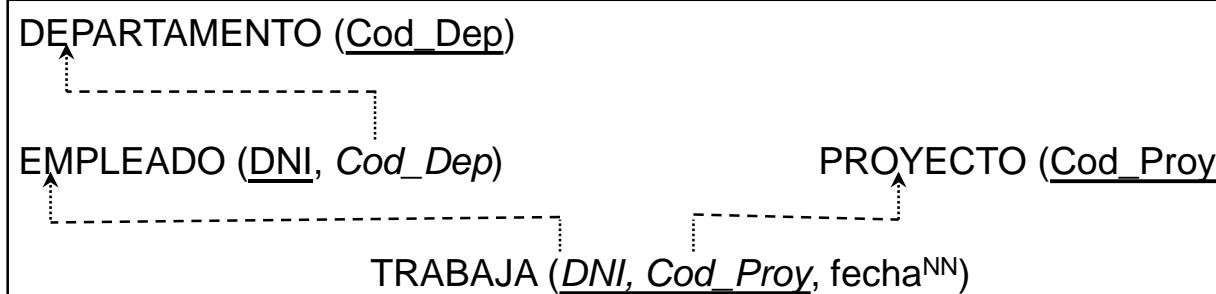
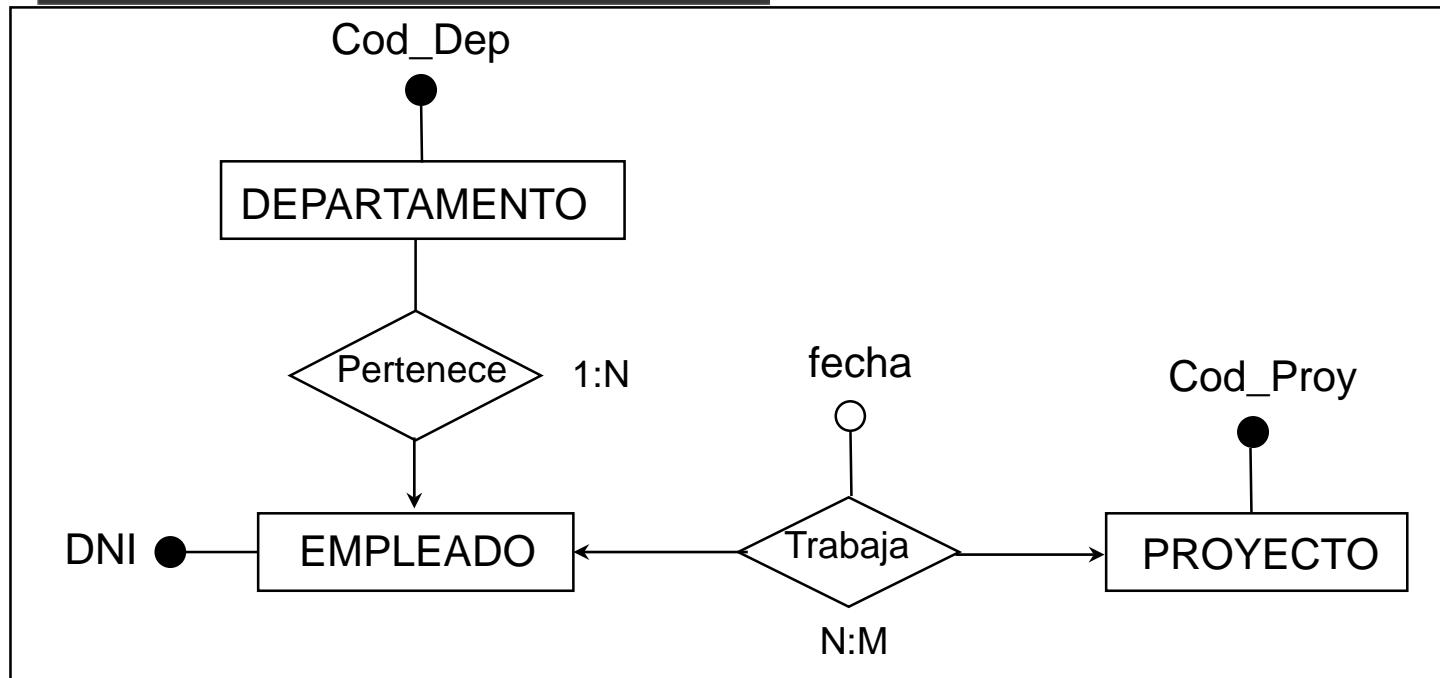
Ejemplo



Diseño de Bases de Datos Relacionales

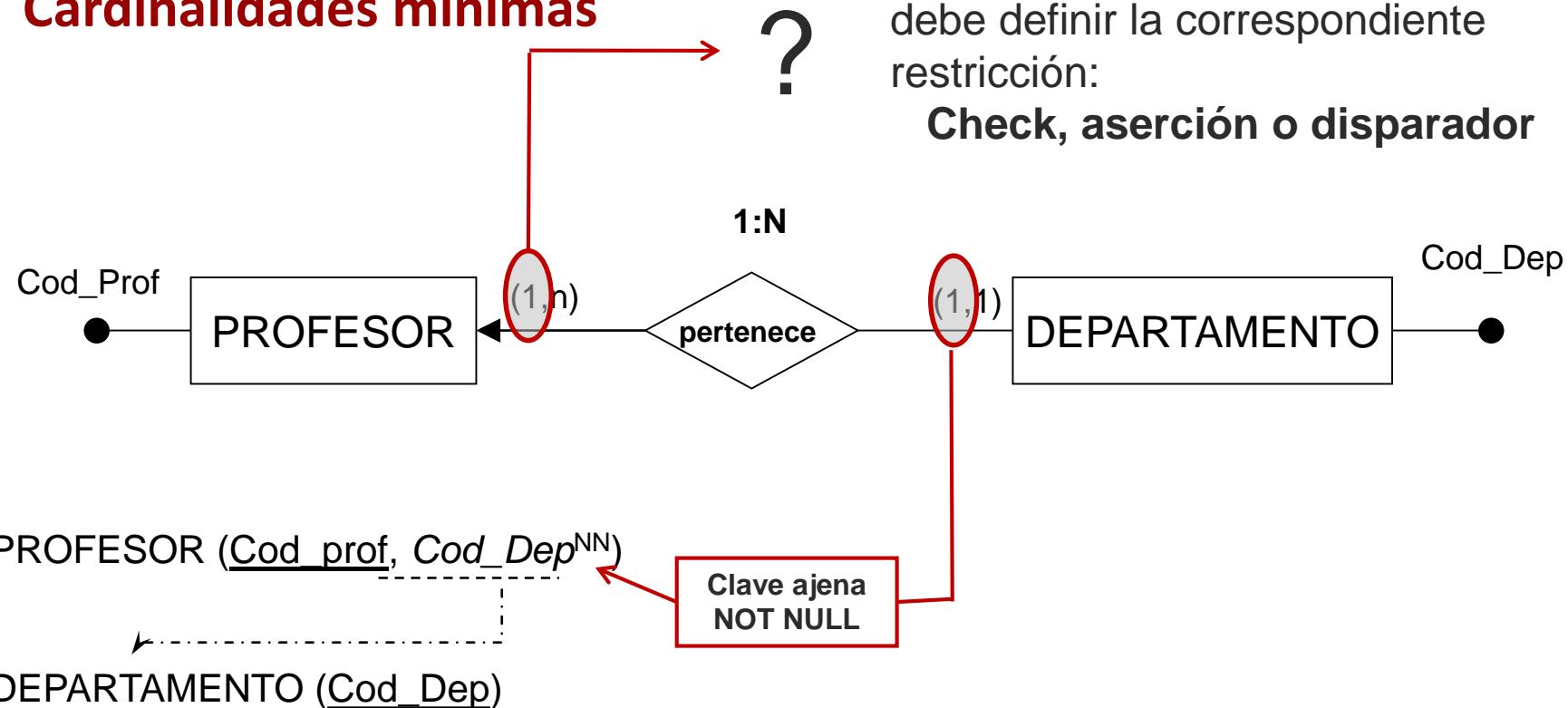
7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Ejercicio básico



7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Cardinalidades mínimas



Interrelaciones

No hay transformación directa. Se debe definir la correspondiente restricción:

Check, aserción o disparador

Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Interrelaciones

❑ **I:N** \Rightarrow 2 posibilidades:

- A. Propagar los Identificadores Principales en el sentido de la flecha (**regla básica**)
- **B. Transformarlo en una relación**

❑ **Criterios para transformar en relación:**

- Si aparecen **muchos** valores nulos
- Cuando se prevé que dicha interrelación en un futuro se convertirá en una **de tipo N:M**
- Cuando la interrelación tiene **atributos propios** y no deseamos propagarlos (a fin de conservar la semántica)

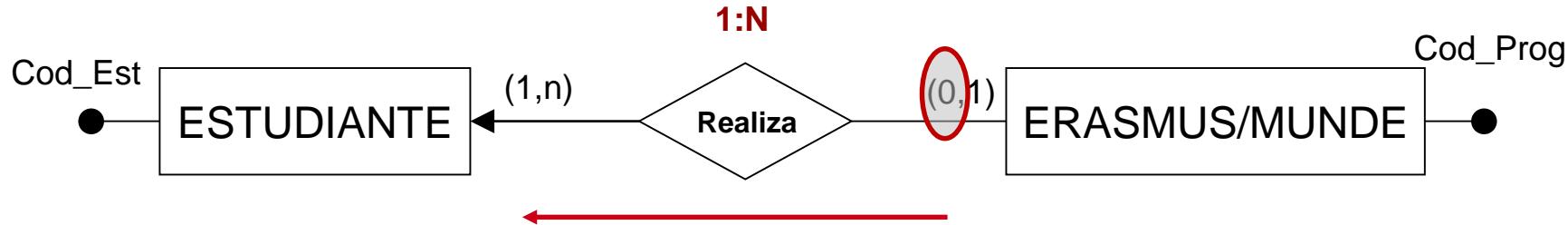
❑ Mecanismos de borrado y actualización (dependiente de la semántica)

Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Interrelaciones

- 1: N : Si aparecen muchos valores nulos



Realiza (Cod_Est, Cod_Prog^{NN})

B:C, M:C

ESTUDIANTE (Cod_Est)

B:C, M:C

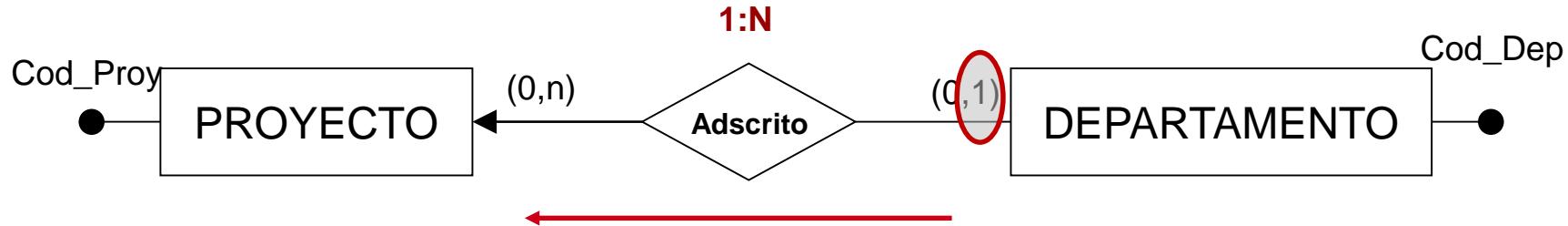
ERAS_MUN (Cod_Prog)

Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Interrelaciones

- 1: N : Cuando se prevé que dicha interrelación en un futuro se convertirá en una **de tipo N:M**



ADSCRITO (Cod_Proy Cod_Dep^{NN})

B:C, M:C

PROYECTO (Cod_Proy)

B:C, M:C

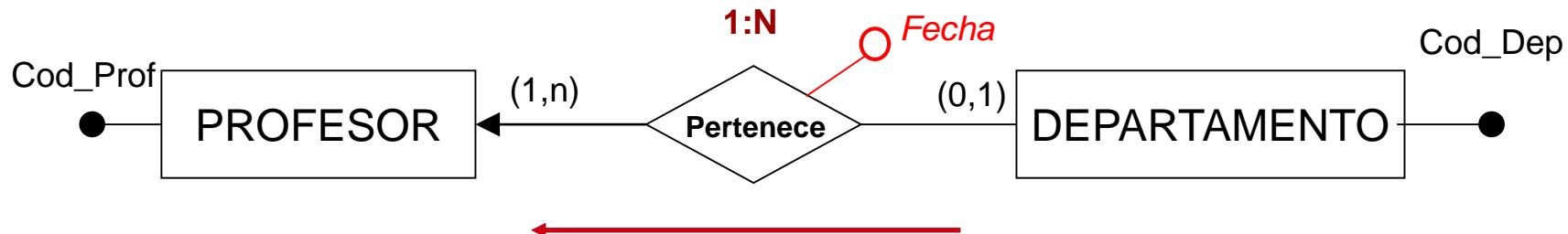
DEPARTAMENTO (Cod_Dep)

Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Interrelaciones

- 1: N : Cuando la interrelación tiene **atributos propios** y no deseamos propagarlos (a fin de conservar la semántica)



Pertenece (Cod_Prof, **Cod_Dep^{NN}**, Fecha^{NN})

B:C, M:C

PROFESOR (Cod_Prof)

B:C, M:C

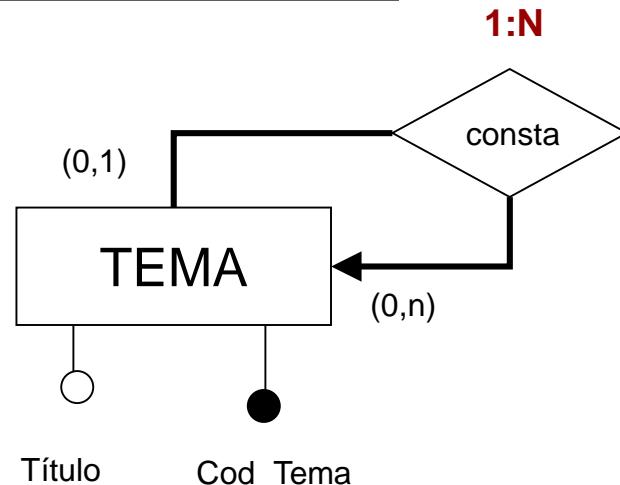
DEPARTAMENTO (Cod_Dep)

Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Ejemplo

Ejemplo



37

Solución A:

B:N, M:C

→ TEMA (Cod_tema, Título^{NN}, Cod_tema_sup*)

TEMA

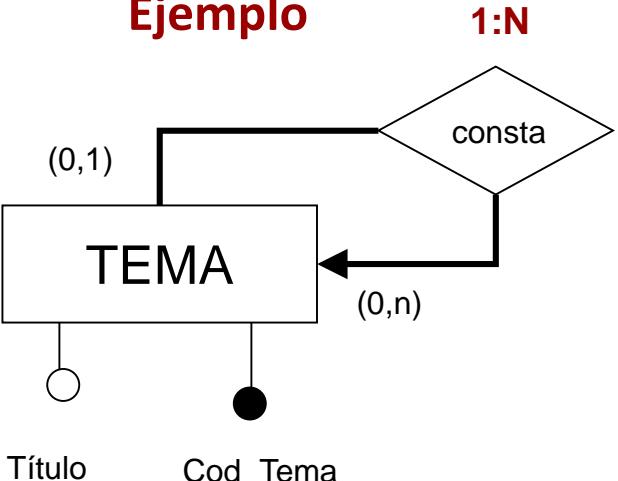
Cod_Tema	Titulo	Cod_Tema_Sup
1	Diseño Lógico	NULL
1.1	Introducción	1
1.2	Etapas del Diseño Lógico	1
1.3	Reglas de Transformación	1
1.3.1	Modelo Básico	1.3
1.3.2	Extensiones	1.3

Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Ejemplo

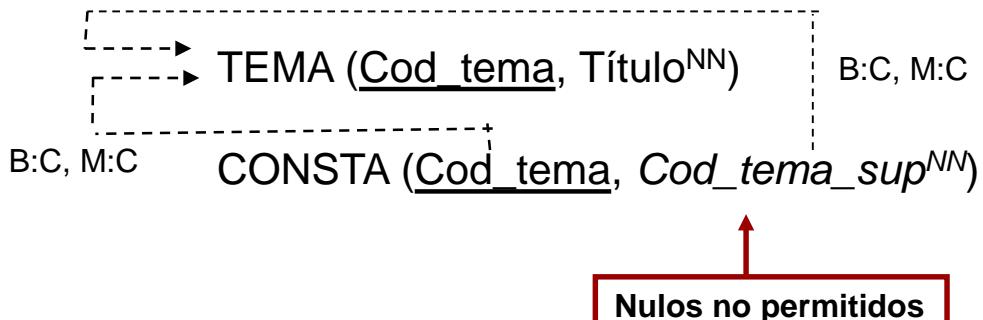
Ejemplo



Cod_Tema	Titulo
1	Diseño Lógico
1.1	Introducción
1.2	Etapas del Diseño Lógico
1.3	Reglas de Transformación
1.3.1	Modelo Básico
1.3.2	Extensões

Cod_Tema	Cod_Tema_Sup
1.1	1
1.2	1
1.3	1
1.3.1	1.3
1.3.2	1.3

Solución B:



```

CREATE TABLE Consta (
    Cod_Tema Codigos,
    Cod_Tema_Sup Codigos,
    PRIMARY KEY (Cod_Tema),
    FOREIGN KEY (Cod_Tema) REFERENCES Tema
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (Cod_Tema_Sup) REFERENCES Tema
        ON DELETE CASCADE
        ON UPDATE CASCADE);
    
```

Ejercicio I a

7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Interrelaciones

- 1:1 \Rightarrow Caso particular de N:M o 1:N
 - Propagar Identificador Principal (incluso en los dos sentidos)
 - Transformarlo en una relación
- Criterios:
 - A. (0,1), (0,1) \Rightarrow **relación**, evitando valores nulos
 - B. (0,1), (1,1) \Rightarrow **propagar la clave**, evitando así valores nulos y
captando más semántica (al recoger la cardinalidad mínima 1)
 - ←
 - C. (1,1), (1,1) \Rightarrow **propagar la clave**, o incluso las dos (por motivos de
eficiencia)
 - ← →

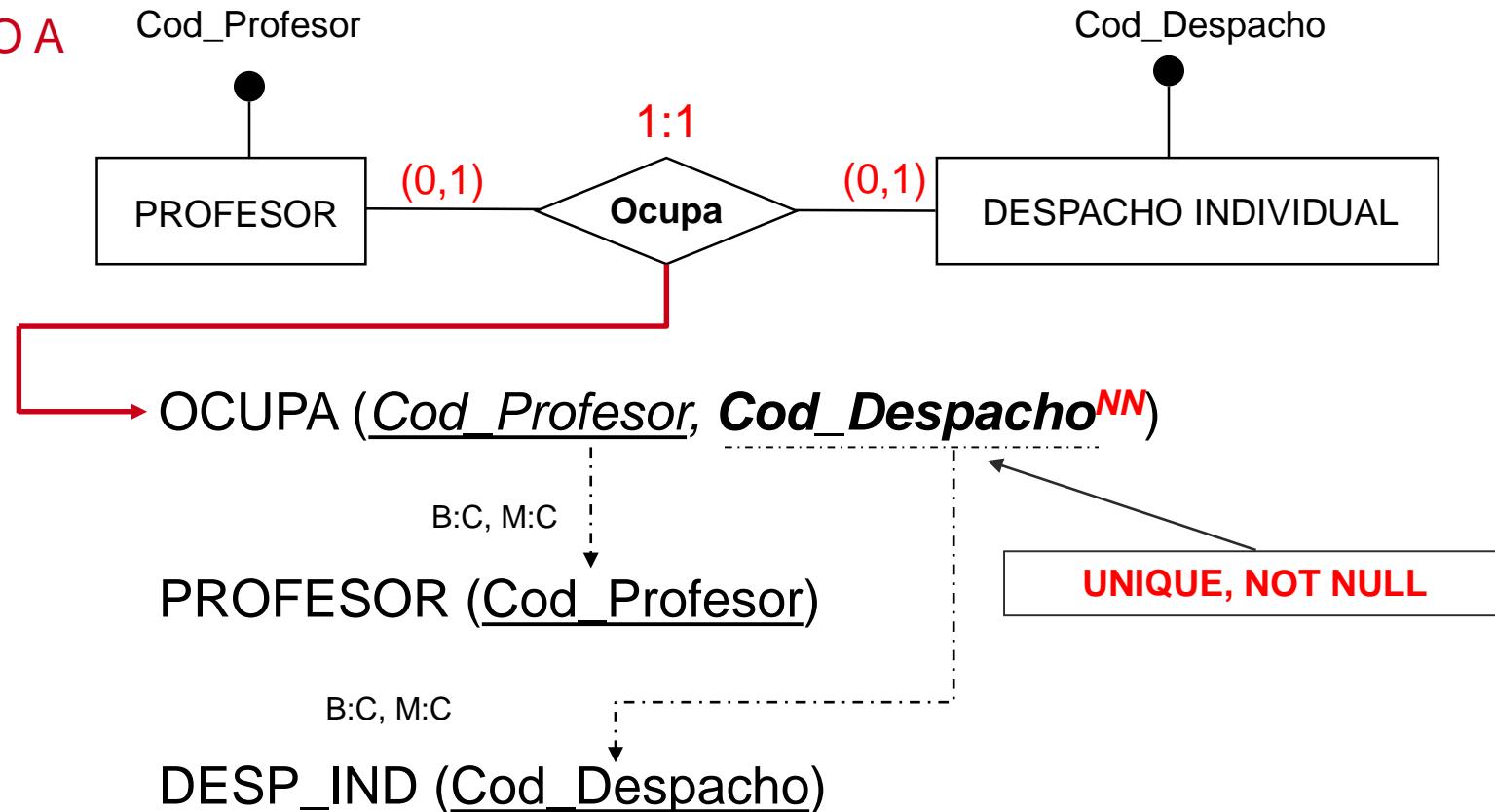
Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Interrelaciones

Ejemplo

CASO A



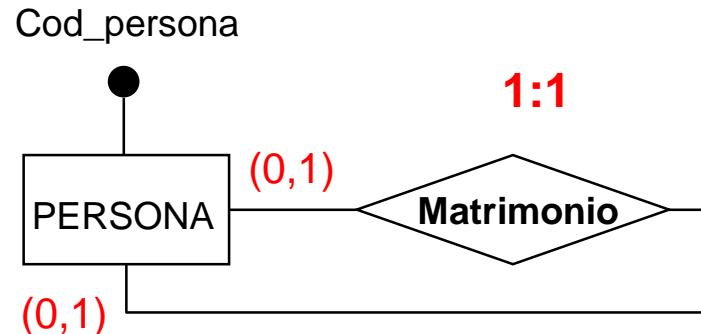
Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Interrelaciones

Ejemplo

CASO A
(reflexiva)



MATRIMONIO (Cod_pers1, Cod_pers2^{NN})

PERSONA (Cod_persona)

UNIQUE, NOT NULL

B:C, M:C

B:C, M:C

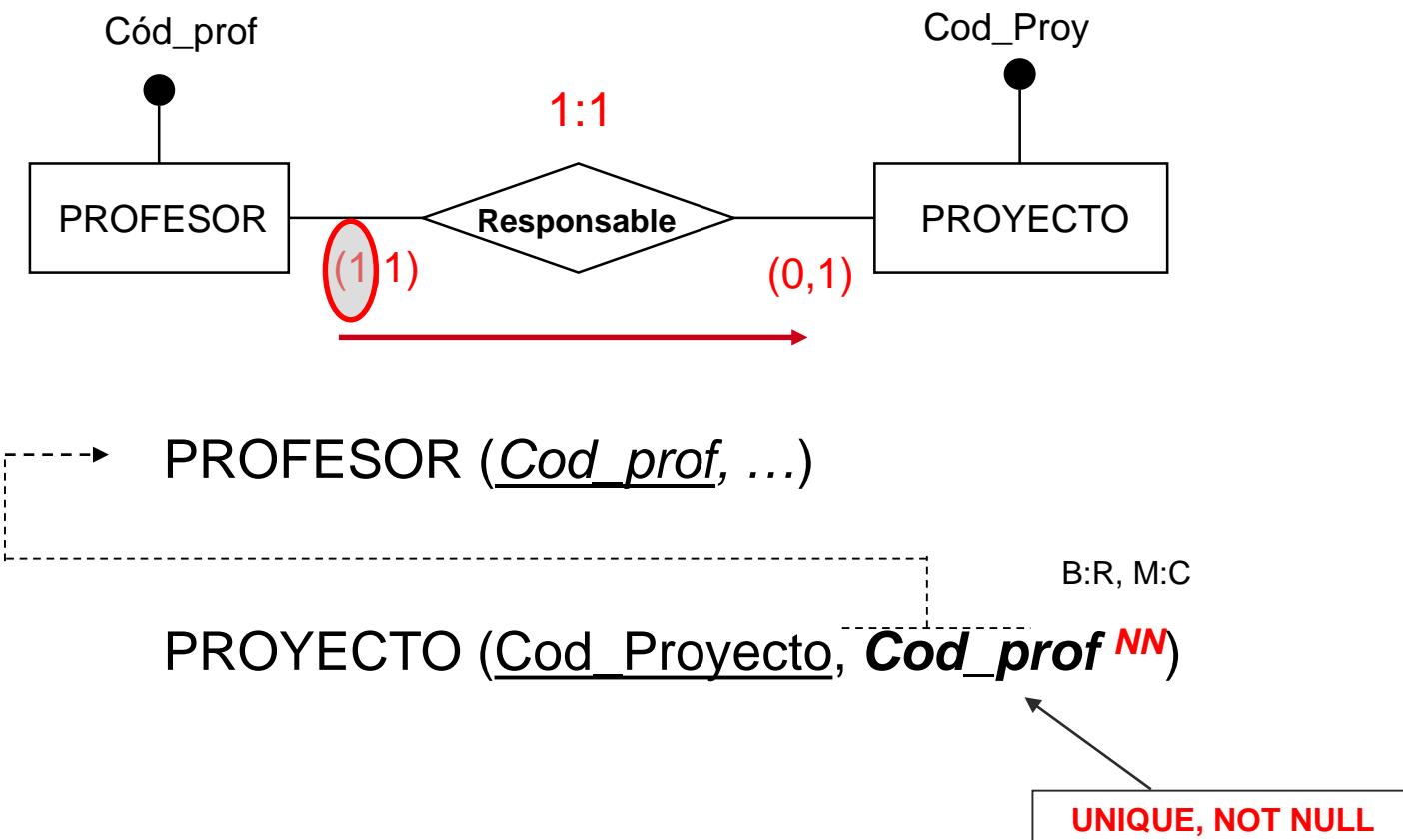
Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Interrelaciones

Ejemplo

CASO B



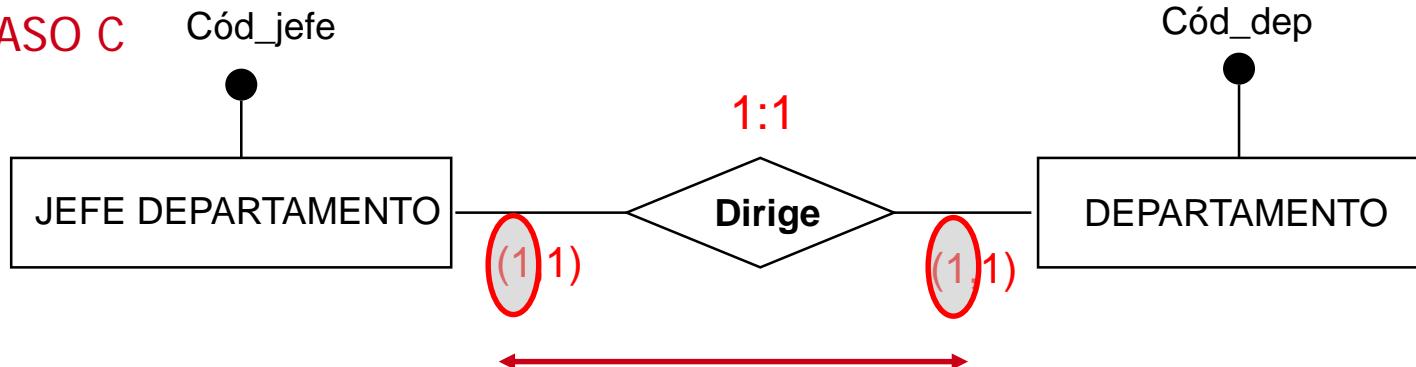
Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Interrelaciones

Ejemplo

CASO C



→ JEFE_DEPARTAMENTO (Cód_jefe)

DEPARTAMENTO (Cód_dep, **Cód_jefe^{NN}**)

B:R, M:C

UNIQUE, NOT NULL

7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Interrelaciones: Dependencias en Existencia y en Identificación

Dependencias en identificación y en existencia

- No se recoge directamente en el Modelo Lógico Estándar.



Utilizaremos los siguientes mecanismos:

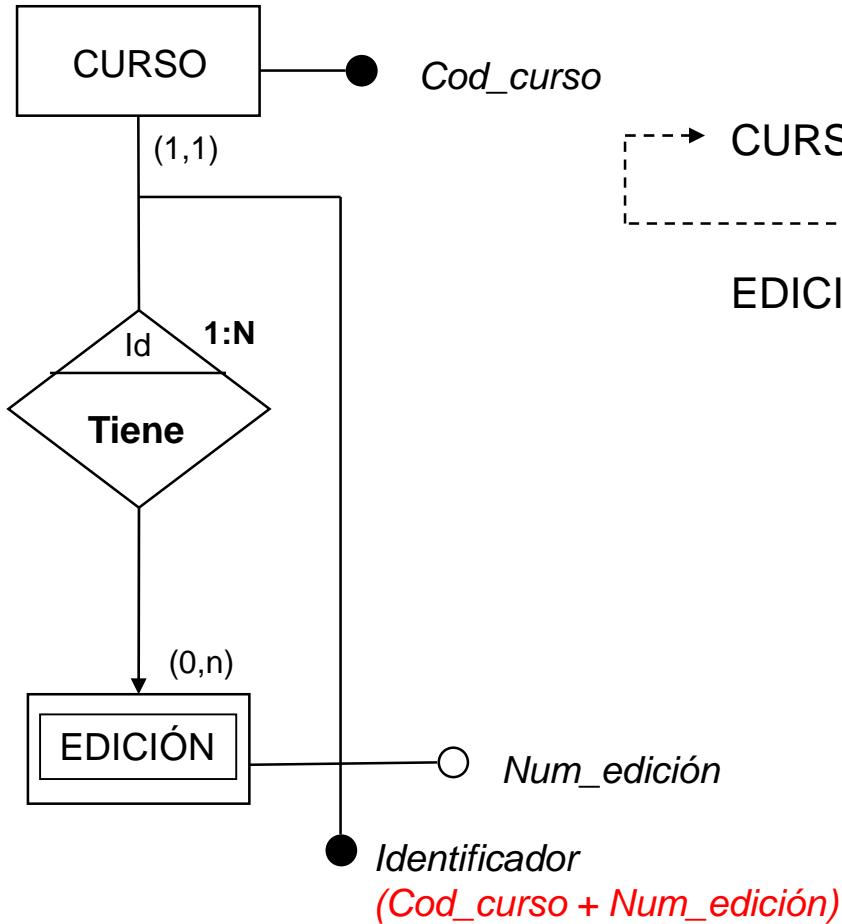
- **Dependencia en Existencia**
 - Propagación de clave ajena con nulos no permitidos.
 - Modificación y borrado en cascada.
- **Dependencia en Identificación**
 - Además de lo anterior, **se concatenan las claves**.

Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar

7.2.1 Reglas de Transformación

Interrelaciones: Dependencias en Existencia y en Identificación



B:C, M:C

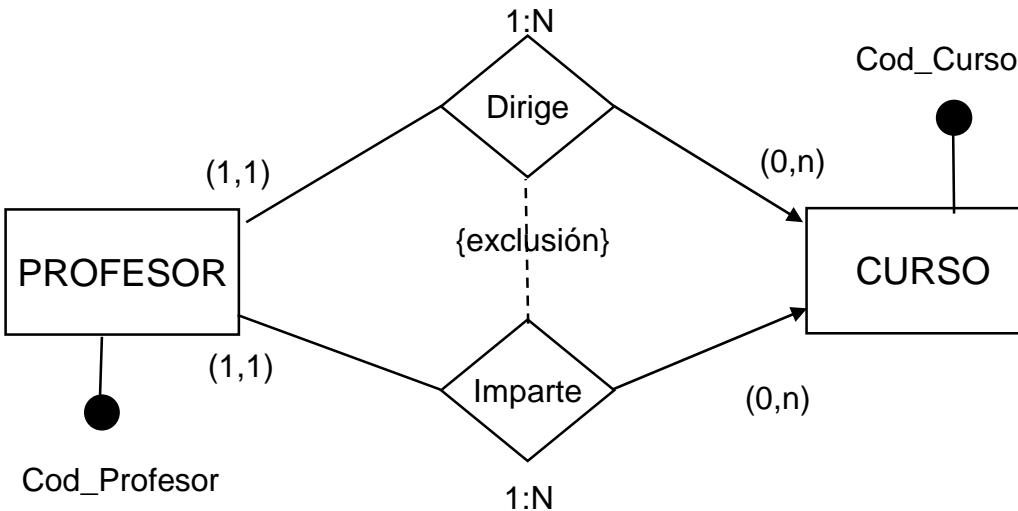
- Nulos no permitidos (sólo en Existencia)
- Forma parte de la clave (en Identificación)

Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Restricciones de Interrelaciones

- Restricciones de interrelaciones (exclusión, inclusión)
 - Se deberán definir las restricciones (Checks, aserciones) pertinentes en cada caso.



CREATE TABLE Curso (
Cod_Curso Codigos_Cursos,
Cod_prof_dirige Cods_Profesores NOT NULL,
Cod_prof_imparte Cods_Profesores NOT NULL,
PRIMARY KEY (Cod_Curso)
FOREIGN KEY (Cod_prof_dirige)
REFERENCES Profesor
ON UPDATE CASCADE
FOREIGN KEY (Cod_Prof_imparte)
REFERENCES Profesor
ON UPDATE CASCADE
CHECK (Cod_prof_dirige <> Cod_prof_imparte)
)

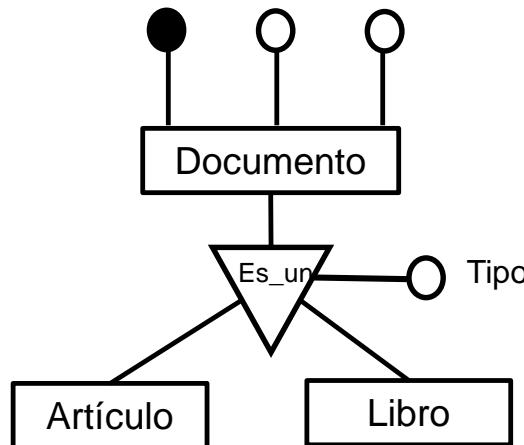
Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación

Generalizaciones

Transformación de generalizaciones

Código Título Idioma



Opción a: una sola tabla
Documento (Código, Titulo, Idioma, Tipo)

Opción b: tres tablas
Documento (Código, Titulo, Idioma)
Artículo (Código) Libro (Código)

Opción c: dos tablas
Artículo (Código, Título, Idioma)
Libro (Código, Título, Idioma)



Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar
7.2.1 Reglas de Transformación

Generalizaciones

Tipos y subtipos

No se pueden representar explícitamente en el Modelo Relacional. Existen varios tipos de transformación:

50

Opción a:

- Englobar tipos y subtipos en **una única relación**. **Subtipos similares en atributos e interrelaciones.**
- Atributo discriminante: estudiar su comportamiento en caso de **totalidad y exclusividad**.
- Mayor eficiencia.

Generalizaciones

Opción b:

- Tantas **relaciones como tipos y subtipos**. Múltiples atributos e interrelaciones.
- Estudio de las restricciones.
- Menos eficiente que opción a), pero conserva **más la semántica**.

51

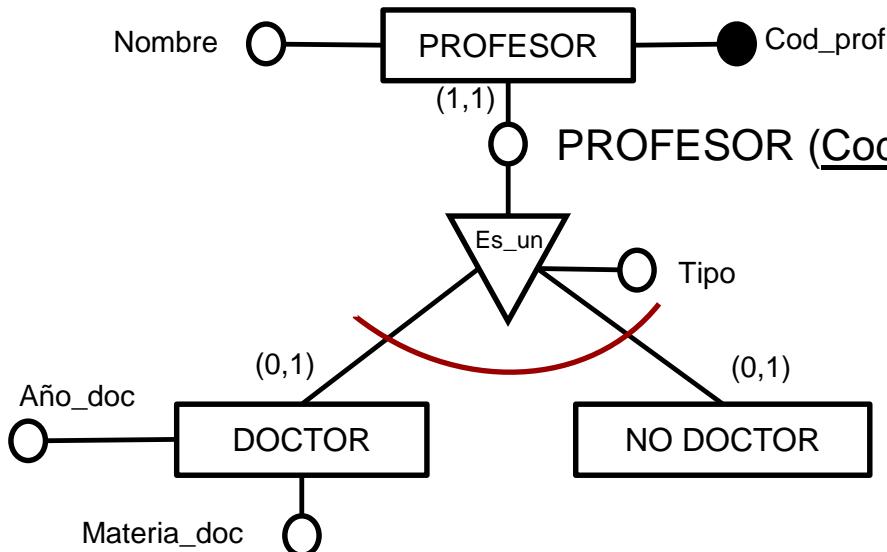
Opción c:

- Relaciones distintas por **cada subtipo** que incluya los atributos propios y comunes.
- Estudio de restricciones.
- Mayor eficiencia ante determinadas consultas (por ej., en el caso de muchos accesos a atributos comunes). Posibles redundancias en caso de solapamiento. ¿En el caso de jerarquía parcial?

Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar

7.2.1 Reglas de Transformación



Generalizaciones

Ejemplo

Solución a)

52

¿Acepta nulos?

Jerarquía Parcial → Acepta Nulos
Jerarquía Total → No Acepta Nulos

```

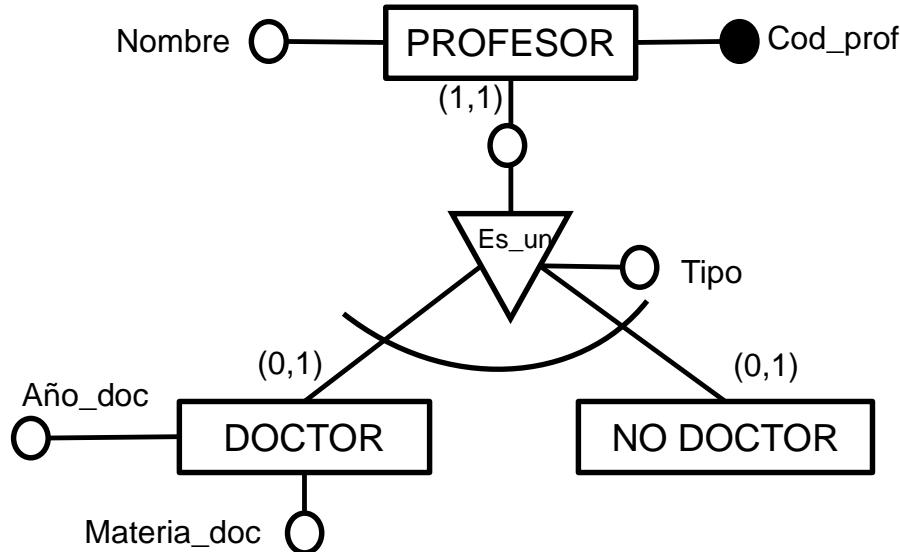
CHECK (( Tipo = 'NO_DOCTOR'
        AND Año_Doc IS NULL
        AND Materia_Doc IS NULL))
      OR (Tipo = 'DOCTOR'
        AND Año_doc IS NOT NULL
        AND Materia IS NOT NULL);
  
```

¿Y si hay solapamiento?

Tipo → grupo repetitivo → nueva relación
Más eficiente: codificar los posibles subtipos

Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar 7.2.1 Reglas de Transformación



Generalizaciones

Ejemplo

Solución b)

DOCTOR (Cod_Prof, Año_doc^{NN}, Materia_doc^{NN})

PROFESOR (Cod_Prof, Nombre^{NN}, **Tipo^{NN}**)

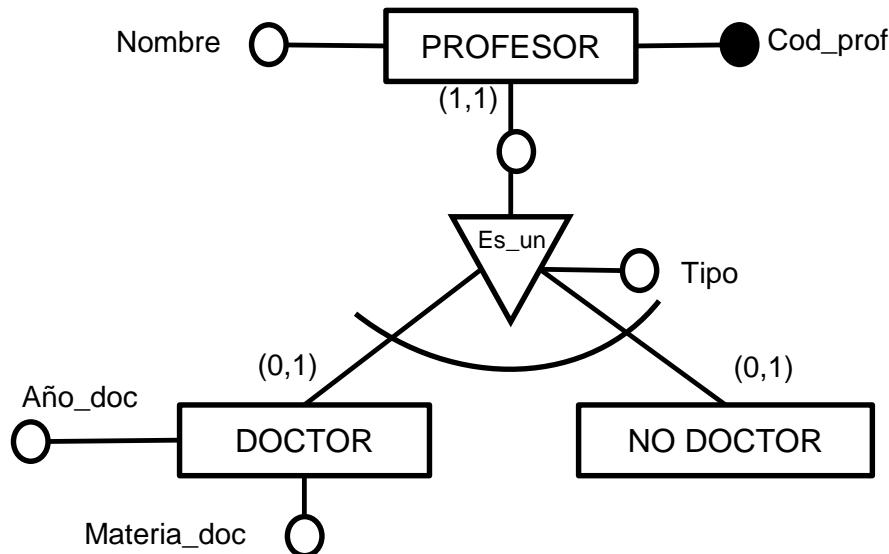
NO_DOCTOR (Cod_Prof)

Diseño de Bases de Datos Relacionales

7.2 Diseño Lógico Estándar

7.2.1 Reglas de Transformación

Generalizaciones



Ejemplo Solución c)

DOCTOR (Cod_prof, Nombre^{NN}, Año_doc^{NN}, Materia_doc^{NN})

NO_DOCTOR (Cod_prof, Nombre^{NN})



Diseño de Bases de Datos Relacionales

Contenido

Tema 7: Diseño de Bases Datos Relacionales

7.1 *Metodología de Diseño de Bases de Datos Relacionales*

7.2 *Diseño Lógico Estándar*

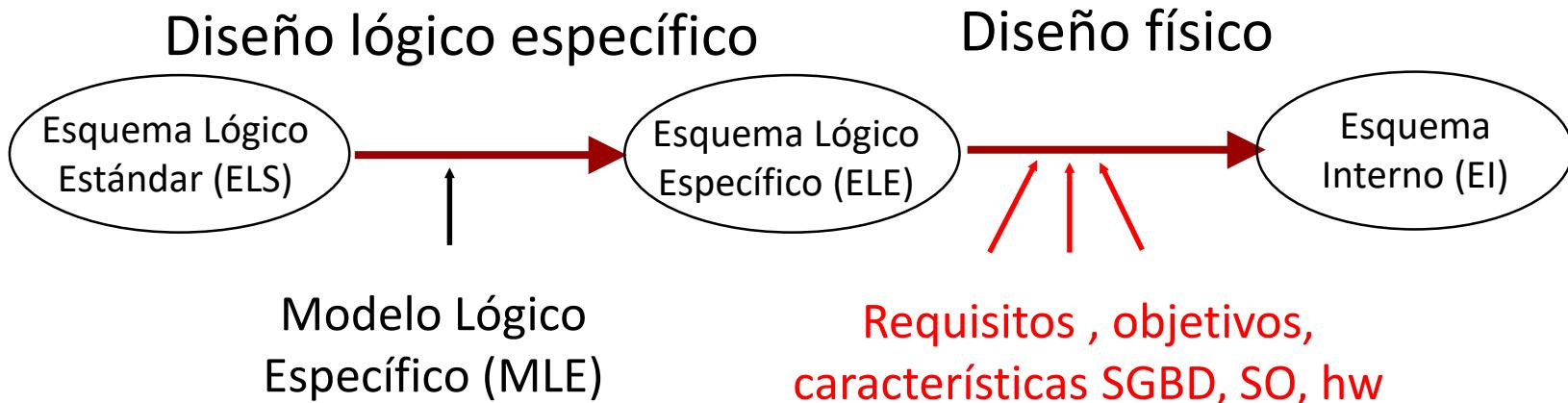
 7.2.1 *Reglas de Transformación*

7.3 *Diseño Lógico Específico y Físico*

7.4 *Teoría de Normalización*

Diseño de Bases de Datos Relacionales

7.3 Diseño Lógico Específico y Físico



- **Obtención del ELE a partir del ELS**, teniendo en cuenta el MLE, descrito en el Lenguaje de Definición de Datos (LDD) **del producto**
- **Estudio del SGBD:**
 - Grado de soporte del Modelo Lógico Estándar (MLS)
 - Adaptación del ELS a características del MLE
 - Definición del ELE en la sintaxis del SGBD



Diseño de Bases de Datos Relacionales

7.3 Diseño Lógico Específico y Físico

Correspondencia entre conceptos del MLS y MLE:

- ❑ Soporte de los conceptos del MLS
- ❑ No soporta ciertos conceptos:
 - Utilizar conceptos propios (tablas, restricciones, disparadores etc.)
 - Restricciones en los programas (Integridad Referencial,...)

Diseño de Bases de Datos Relacionales

7.3 Diseño Lógico Específico y Físico

Ejemplo: Dominios

- Casi ningún producto los soporta
- Posible simulación, tabla, reglas, disparadores

Ejemplo: Dominio de Estados Civiles representado mediante una tabla

ESTADOS CIVILES

EST_CIVIL
S
C
V
D

UPDATE
DELETE
INSERT

}

NO PERMITIDO



Diseño de Bases de Datos Relacionales

7.3 Diseño Lógico Específico y Físico

Objetivos del diseño físico:

- Disminuir los tiempos de respuesta
- Minimizar espacio de almacenamiento
- Evitar las reorganizaciones
- Proporcionar la máxima seguridad
- Optimizar el consumo de recursos

59

Es decir, cumplir los **objetivos del sistema** y conseguir la **optimización del ratio coste/beneficio**.



Diseño de Bases de Datos Relacionales

7.3 Diseño Lógico Específico y Físico

Entradas

- Objetivos del diseño físico, con prioridades y cuantificados
- Esquema lógico
- Recursos hw
- Recursos sw
- Información sobre las aplicaciones

60

Salidas

- Estructura interna
- Especificaciones para el ajuste o *tuning*
- Normas de seguridad



Diseño de Bases de Datos Relacionales

7.3 Diseño Lógico Específico y Físico

Estrategias:

- ❑ El SGBD **impone** una estructura interna
- ❑ El administrador **diseña** la estructura interna
- ❑ El SGBD proporciona un estructura interna a partir de algunos **parámetros modificables**:
 - La BD puede empezar a funcionar de inmediato
 - La eficiencia va aumentando al irse efectuando sucesivos ajustes
 - Se mantiene la independencia físico/lógica



Diseño de Bases de Datos Relacionales

7.3 Diseño Lógico Específico y Físico

El administrador debe decidir entre las opciones disponibles a fin de conseguir un **acceso más eficiente** dados los requisitos concretos del sistema.

62

80%-90% de las manipulaciones sobre la BD las realiza un 10%-20% de las aplicaciones: **consultas frecuentes y prioritarias.**



Diseño de Bases de Datos Relacionales

Contenido

Tema 7: Diseño de Bases Datos Relacionales

7.1 *Metodología de Diseño de Bases de Datos Relacionales*

7.2 *Diseño Lógico Estándar*

 7.2.1 *Reglas de Transformación*

7.3 *Diseño Lógico Específico y Físico*

7.4 **Teoría de Normalización**