Unidad de Trabajo 2
Diseño conceptual y lógico de bases de datos

El modelo relacional

IES Palomeras Vallecas

Curso 2020/2021

Profesor: Alberto Ruiz

Introducción

- En el proceso de diseño de bases de datos, tenemos ya un diseño conceptual (modelo Entidad/Relación)
- Ahora queremos un diseño lógico, más técnico: estudiaremos el modelo relacional, que es el más extendido
- Relación se refiere aquí a una relación de elementos (tablas).
  - No tiene nada que ver con “Entidad/Relación”

Estructura del modelo

Elementos básicos

- **Relación** *(no confundir con Relación de modelo E/R)*

	Inglés	Castellano
Modelo Entidad/Relación	Relationship	Relación
Modelo Relacional	Relation	Relación

  - Es la estructura básica del modelo relacional.
  - Se representan utilizando *tablas*

Atributo 1	Atributo 2	.....	Atributo n
xxxxx	xxxxx	xxxxx	xxxxx
xxxxx	xxxxx	xxxxx	xxxxx
xxxxx	xxxxx	xxxxx	xxxxx

Tupla 1

.....

Tupla N

Estructura del modelo

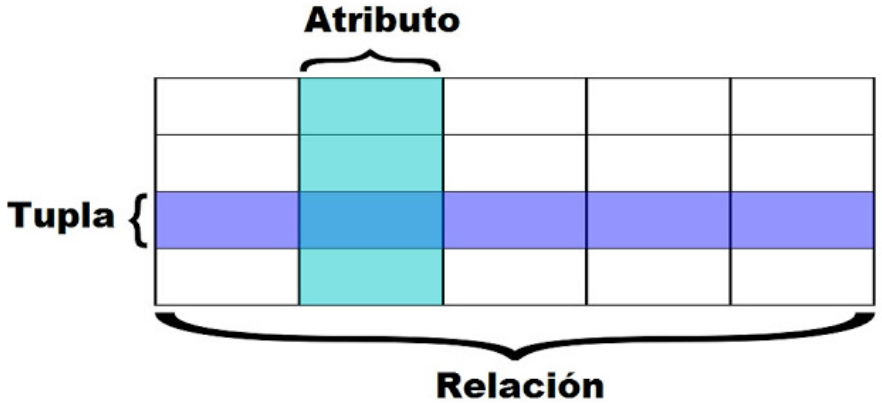
Elementos básicos

- **Tupla**
  - Ocurrencia de la relación (serán las *filas* de las tablas)
- **Atributo**
  - Son las propiedades de la relación (tabla) y se representan mediante *columnas*.
- **Dominio**
  - Conjunto de valores que puede tomar un atributo

Estructura del modelo

Tuplas

- Una tupla representa una ocurrencia de una entidad (“Profesor”) o de una relación (“Imparte”) del modelo Entidad/Relación
- Su representación informal es una fila de tabla.
- Una relación del Modelo Relacional es un conjunto de tuplas, que nosotros vemos como una tabla



Estructura del modelo

Atributos y dominio

- Un atributo representa una propiedad de una relación.
  - Tomará valores dentro de un dominio.

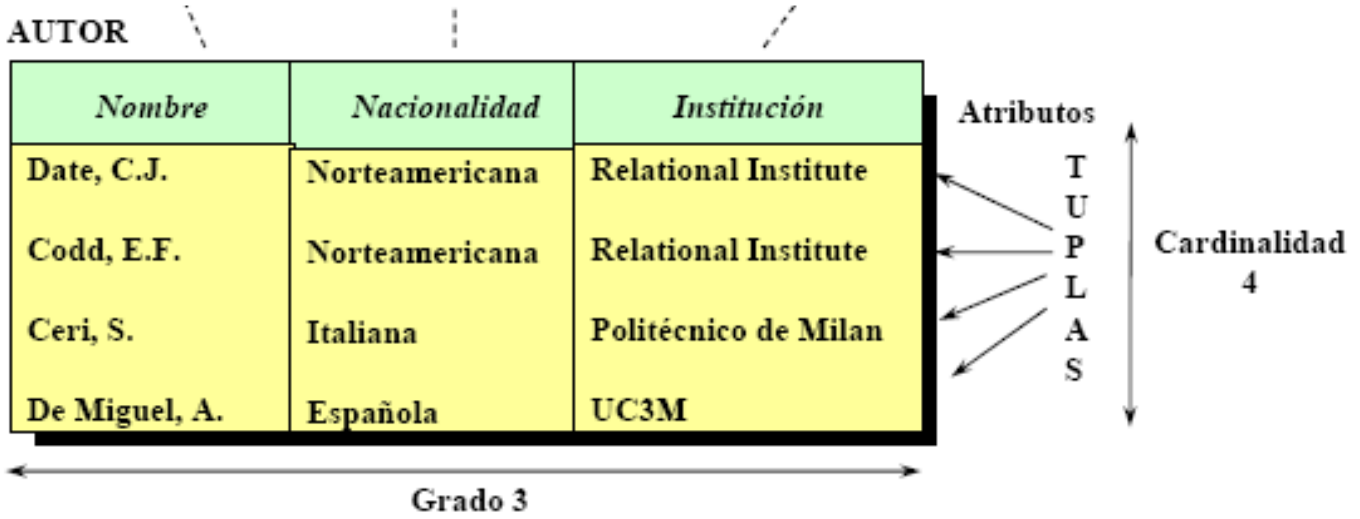
- Dominio: posibles valores de un atributo
  - Puede expresarse de dos formas
    - Indicando sus posibles valores
      - **Ejemplo:** días de la semana = {lunes, martes, miércoles, … sábado, domingo}
    - Indicando su tipo de datos
      - **Ejemplos:** edad: número entero, nombre: texto

- Dominio compuesto
  - El dominio compuesto denominado **Fecha** se construye por agregación de los dominios simples **Día, Mes y Año**.

Estructura del modelo

Cardinalidad y grado

- Cardinalidad
  - Número de tuplas (filas) de la relación
- Grado
  - Número de atributos (columnas)

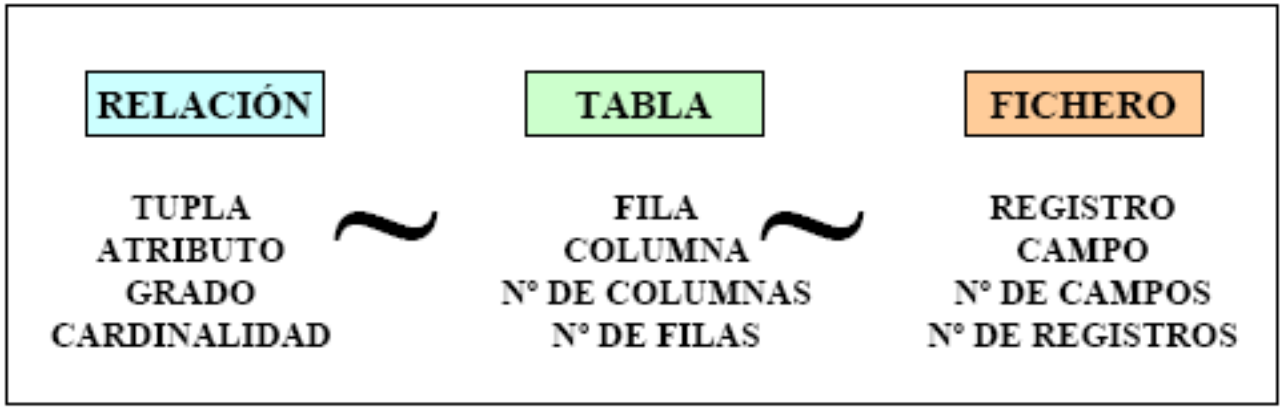


Estructura del modelo

Tabla vs Relación

- Usaremos la palabra “tabla” en lugar de “relación” para evitar confusiones con el modelo E/R, pero…
- **Una relación tiene algunas diferencias:**
  - El orden de las filas y columnas es irrelevante
  - No puede haber dos tuplas iguales (para eso usaremos la clave primaria)
  - El atributo que forme parte de la clave primaria no podrá estar vacío
  - Cada atributo sólo puede tener un valor
    - No se admite: “Teléfono”→5551532, 91353442, 5553243

Relaciones vs Tablas



Modelo Relacional (teoría)

SGBD Relacionales (implementación)

Sistemas de Ficheros Clásicos

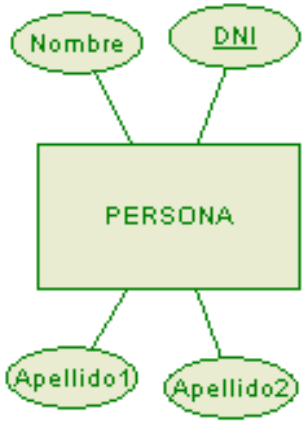
Paso del modelo E/R al relacional

- **Proceso de diseño**
  - A continuación se explicarán las reglas para transformar un diagrama Entidad/Relación en un diseño de modelo relacional. Aprenderemos a transformar:
    - Entidades
    - Relaciones N:M
    - Relaciones 1:N
    - Relaciones 1:1
    - Relaciones de diferentes grados
    - Atributos multivaluados
    - Jerarquías (generalizaciones)
    - Dependencias en existencia y en identificación

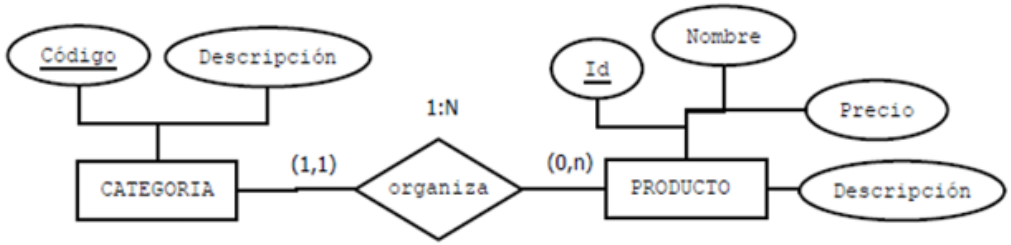
Paso del modelo E/R al relacional

Entidades

- Toda entidad se transforma en una tabla



PERSONA (DNI, Nombre, Apellido1, Apellido2)



CATEGORÍAS (Código, Descripción)

PRODUCTOS(Id, Nombre, Precio, Descripción)

Si lo prefieres, en las tablas sí puedes usar el plural (Productos en vez de Producto)

En el diagrama E/R es importante destacar que representamos "un" producto (los plurales vienen indicados por las cardinalidades)

Paso del modelo E/R al relacional

Atributos

- Los atributos opcionales se marcan con un *
  - Esto puede ser difícil de recordar porque en los formularios web, el * indica lo contrario (“obligatorio”)

CLIENTE (DNI, Nombre, Apellido1, Apellido2*, Dirección*, Teléfono*, Email)

PROFESOR (NIF, Nombre, Apellido, Dirección*)

ESTUDIANTE (Nº de Matrícula, Nombre, Apellido)

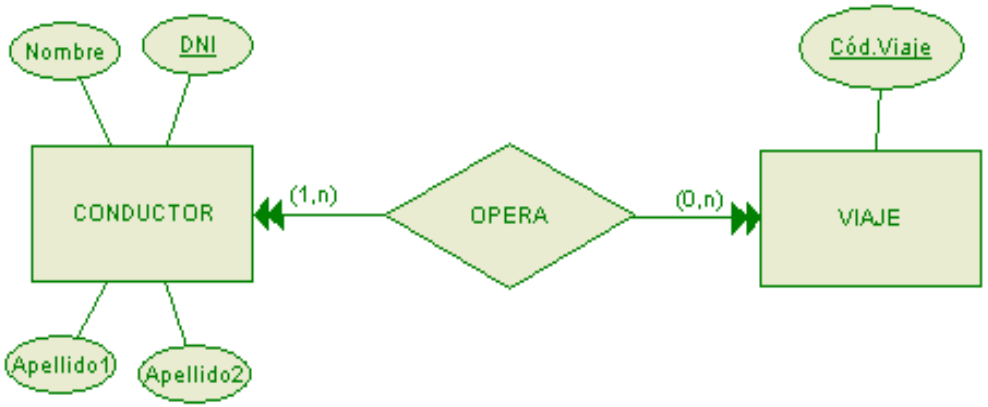
COCHE (Matrícula, Marca, Modelo, Color*)

EJEMPLAR (ISBN, nº Ejemplar, nº de páginas*)

Paso del modelo E/R al relacional

Relaciones N:M

- Toda relación n:m se transforma en una tabla, en la que la clave principal estará compuesta por la unión de las claves principales de ambas entidades



CONDUCTOR (DNI, Nombre, Apellido1, Apellido2)

OPERA (DNI_Conductor, Cód. Viaje)

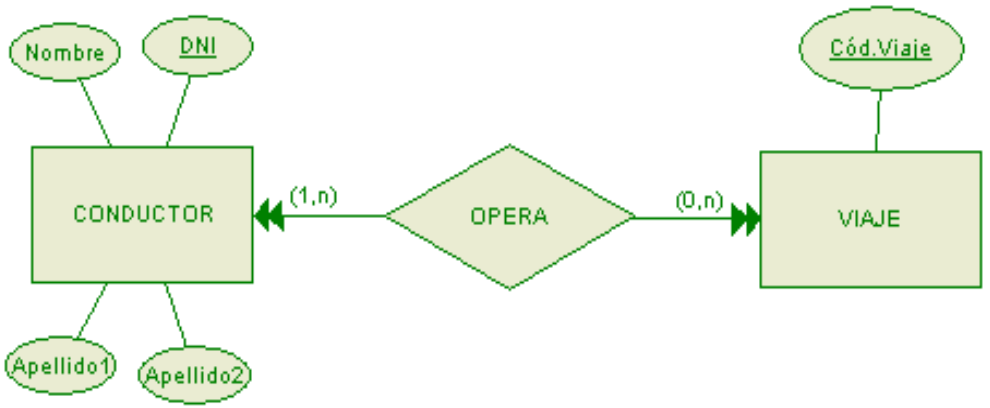
VIAJE (Cód. Viaje)

Cada conductor y cada viaje pueden aparecer muchas veces en la tabla, por eso sus PK no sirven por sí solas como PK de la nueva tabla
La solución es utilizar la combinación de sus PK

Paso del modelo E/R al relacional

Comprendiendo la clave ajena

- Se dice que DNI_Conductor es **clave ajena**, ya que es clave primaria en otro sitio (concretamente en la tabla CONDUCTOR). Representamos las claves ajenas con una flecha que referencia al lugar en el que es clave primaria
  - "Cód. Viaje" también es clave ajena, ya que es PK en "VIAJE"



CONDUCTOR (DNI, Nombre, Apellido1, Apellido2)

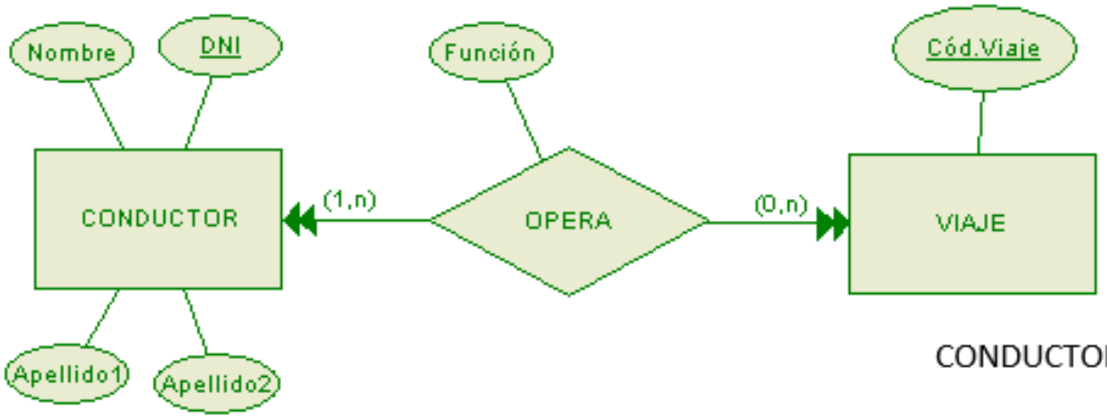
OPERA (DNI_Conductor, Cód. Viaje)

VIAJE (Cód. Viaje)

Paso del modelo E/R al relacional

Relaciones N:M

- Paso de los atributos
  - El atributo de la relación se añade como atributo de la tabla derivada de esa relación.



CONDUCTOR (DNI, Nombre, Apellido1, Apellido2)

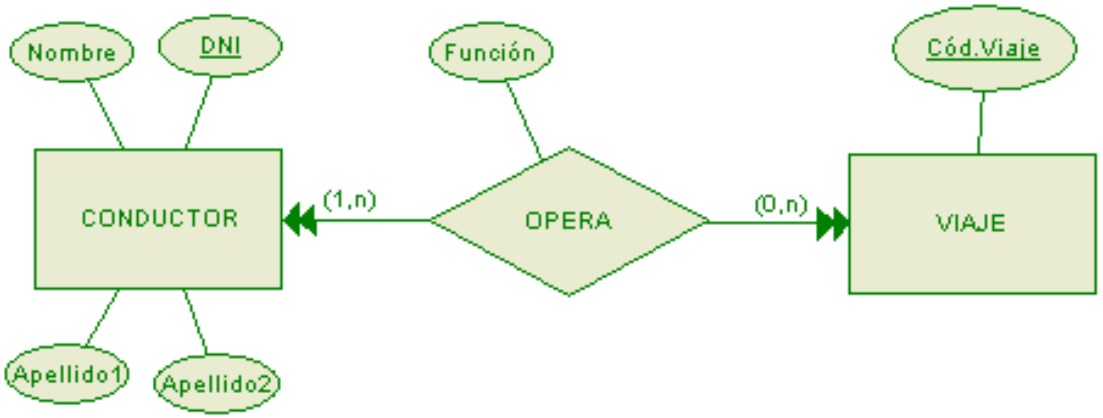
OPERA (DNI_Conductor, Cód.Viaje, Función)

VIAJE (Cód.Viaje)

Paso del modelo E/R al relacional

Relaciones N:M

- Paso de la cardinalidad mínima
  - "Un viaje es operado al menos por un conductor"

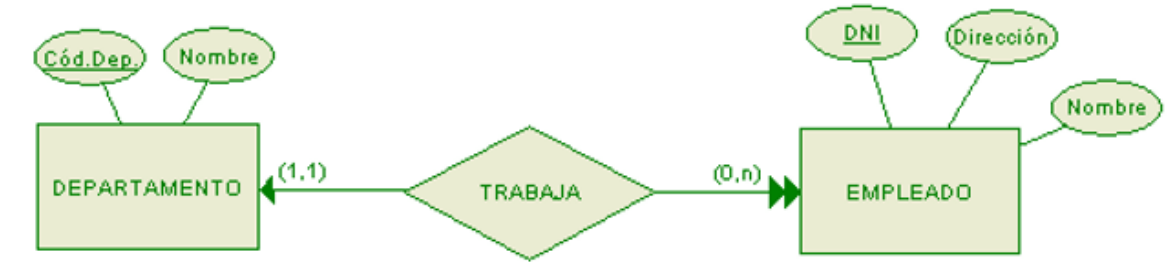


  - Al pasar del modelo Entidad-Relación al modelo relacional **se pierde esta restricción**
  - La comprobación de que un viaje sea operado al menos por un conductor se tendrá que hacer posteriormente con reglas de comprobación (se estudiará más adelante)

Paso del modelo E/R al relacional

Relaciones 1:N

- En las relaciones 1:n no es necesario generar una nueva tabla.
  - Se propaga la clave primaria desde la entidad de cardinalidad 1 hacia la de entidad n.
  - Truco para recordar: se añade un **n**uevo atributo en la entidad de la **n**
- La clave propagada se convierte en clave ajena
- El nombre de la relación se pierde



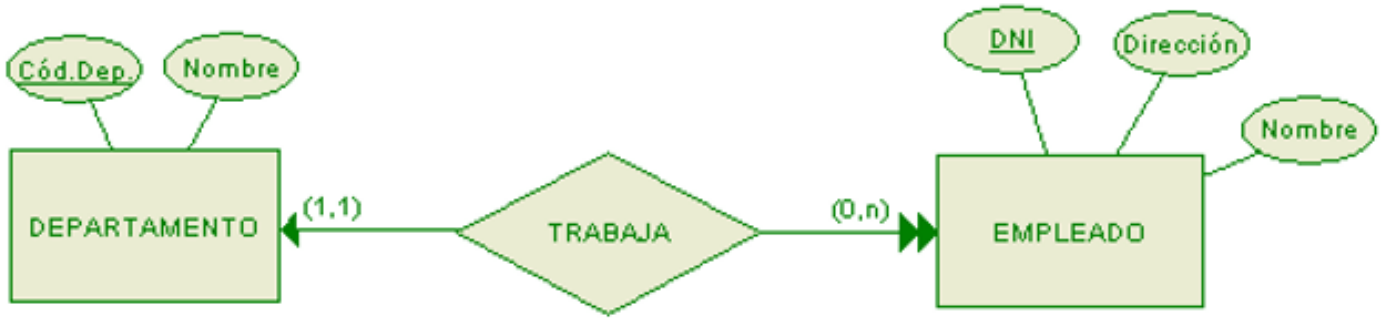
DEPARTAMENTO (Cód.Dep, Nombre)

EMPLEADO (DNI, Dirección, Nombre, Cód.Dep)

Paso del modelo E/R al relacional

Relaciones 1:N

- Si lo hiciésemos al revés (de la entidad de n hacia la de 1) para cada departamento sólo podríamos almacenar un solo DNI de empleado



DEPARTAMENTO (Cód.Dep, Nombre)

EMPLEADO (DNI, Dirección, Nombre, Cód.Dep)

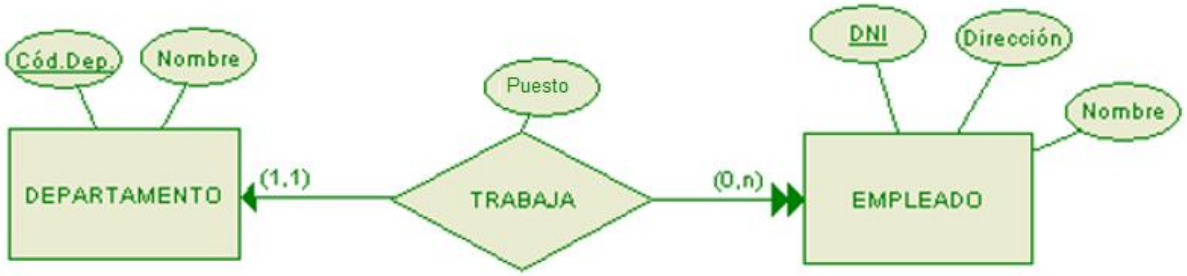
DEPARTAMENTO (Cód.Dep, Nombre, DNI_Empleado)

EMPLEADO (DNI, Dirección, Nombre)

Paso del modelo E/R al relacional

Relaciones 1:N

- Si la relación tiene atributos, hay dos posibilidades:



1. Añadir el atributo a la entidad del lado N:

DEPARTAMENTO (Cód.Dep, Nombre)

EMPLEADO (DNI, Dirección, Nombre, Cód.Dep, **Puesto**)

2. Convertir la relación en una tabla, cuya PK será la de la entidad del lado N:

DEPARTAMENTO (Cód.Dep, Nombre)

TRABAJA (DNI_Empleado, Cód. Departamento, Puesto)

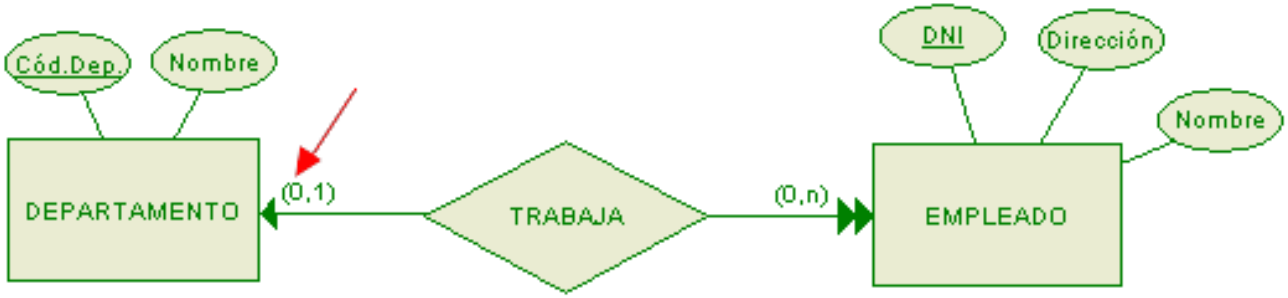
EMPLEADO (DNI, Dirección, Nombre)

A diferencia de las relaciones N:M, aquí cada empleado trabaja sólo en 1 departamento, por eso su DNI aparecerá 1 vez en la tabla TRABAJA y sirve como PK

Paso del modelo E/R al relacional

Relaciones 1:N

- Paso de la cardinalidad mínima en relaciones 1:N
  - Se pierde la distinción, al igual que en las relaciones N:M (estudiaremos mecanismos para expresar esta restricción)



DEPARTAMENTO (Cód.Dep, Nombre)

EMPLEADO (DNI, Dirección, Nombre, Cód.Dep *)

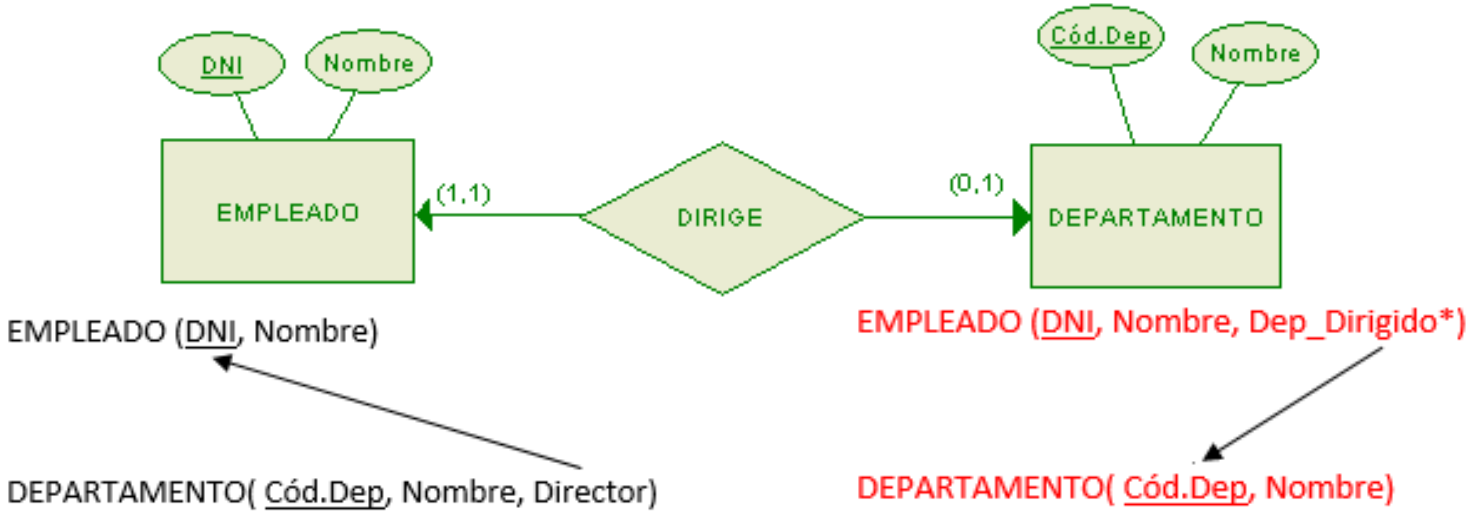
*Recuerda: * indica que el campo puede estar en blanco: Hay empleados que no tienen departamento asociado*

Si pensamos que la relación se va a dar en muy pocos casos, podemos generar una nueva tabla TRABAJA
En este ejemplo, "Cód.Dep" estará casi siempre vacío si los empleados no suelen asociarse a departamentos.

Paso del modelo E/R al relacional

Relaciones 1:1 (caso particular de 1:N)

- Puedes elegir hacia qué lado propagar, pero a veces es mejor un lado que otro:



- La solución de la derecha no es incorrecta, mientras se ponga el * ya que en este caso hay empleados que no dirigen deparamentos, pero es peor:
- ¿Cuántos empleados son directores de un departamento? Muy pocos.
  Hemos creado un campo en la tabla EMPLEADO que casi siempre estará vacío.

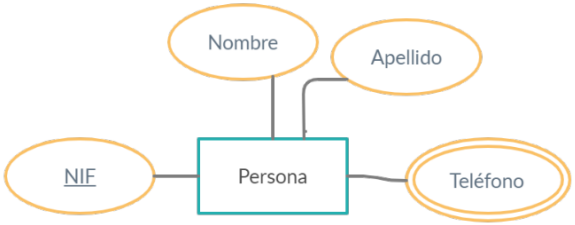
Paso del modelo E/R al relacional

Atributos multivaluados y derivados

Los atributos **multivaluados** generan una nueva tabla con el nombre del atributo:

PERSONA (NIF, Nombre, Apellido)

TELÉFONO (NIF_Cliente, Número)



- La nueva tabla tendrá los siguientes atributos:
  1. El/Los atributos de la clave primaria de la tabla correspondiente a la entidad
  2. El atributo multivaluado

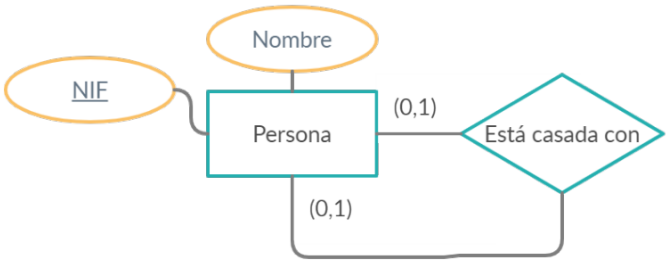
  - Todos ellos formarán la clave primaria de la nueva tabla
  - En el ejemplo, NIF_Persona es **clave ajena**

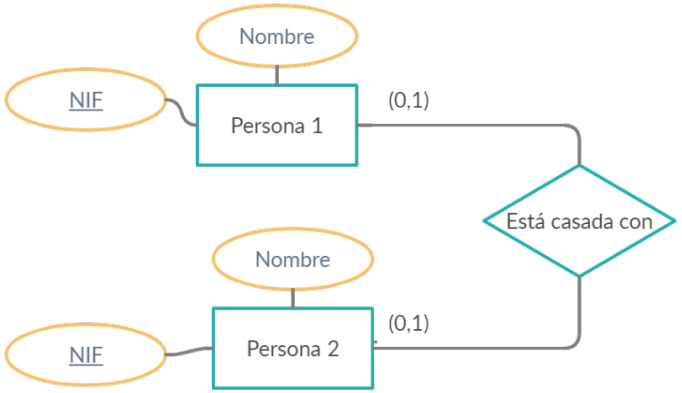
Para los atributos **derivados o calculados** se añadirá posteriormente un disparador (trigger) que calculará su valor. De momento no haremos nada especial

Paso del modelo E/R al relacional

Relaciones de grado 1

- No son un tipo especial de relación: intenta verlas como si participasen dos entidades:





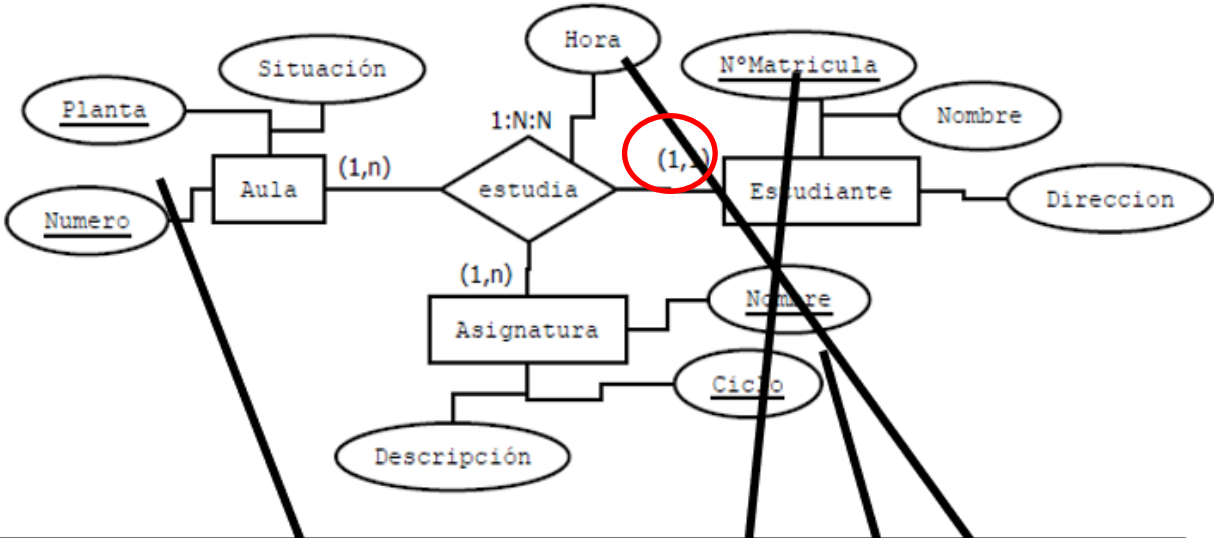
PERSONA (NIF, Nombre, NIF_Cónyuge)

- La única precaución es cambiar el nombre de la clave para que no sea llamen “NIF” las dos
- Utiliza las reglas estudiadas para relaciones 1:1, 1:N y N:M

Paso del modelo E/R al relacional

Relaciones de grado mayor que 2

- Se crea una nueva tabla cuya clave primaria será la unión de las claves de las entidades participantes



AULAS(Numero, Planta, Situación)
ESTUDIANTES(NºMatricula,Nombre,Direccion)
ASIGNATURAS(Nombre, Ciclo,Descripcion)
ESTUDIA (Numero, Planta, NºMatricula, Nombre, Ciclo,Hora)

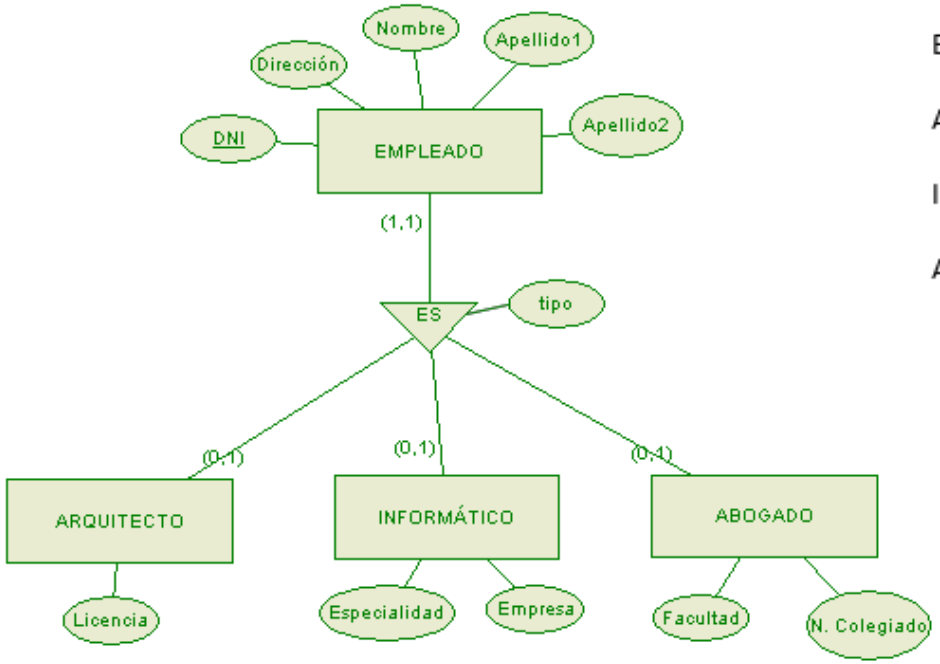
- Si una de las entidades participa con cardinalidad máxima 1, no es necesario que su clave primaria forme parte de la clave primaria de la nueva tabla. Pero por simplicidad, puedes poner siempre todas

Paso del modelo E/R al relacional

Jerarquías

- La solución más equilibrada y recomendada es la siguiente:

1. Se genera una tabla para la entidad padre
2. Se genera una tabla para cada una de las entidades hijas
3. Se propaga la clave principal desde la entidad padre hasta todas las hijas.
4. El atributo “Tipo” se añade a la tabla correspondiente a la entidad padre



EMPLEADO (DNI, Nombre, Apellido1, Apellido2, Dirección, **Tipo**)

ARQUITECTO (DNI, Licencia)

INFORMÁTICO (DNI, Especialidad, Empresa)

ABOGADO (DNI, Facultad, N.Colegiado)

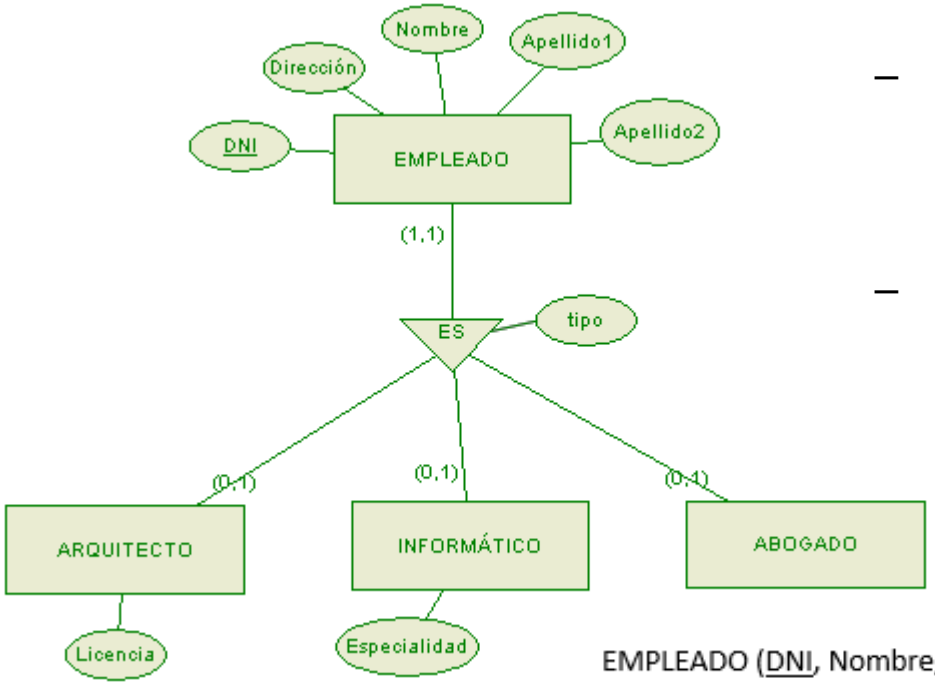
DNI es clave principal para las entidades hijas pero también clave ajena en Empleado: esto permite acceder a los atributos comunes

Paso del modelo E/R al relacional

Jerarquías

- Si vemos que en la jerarquía tiene más peso la entidad padre:

  1. Se genera una tabla para la entidad padre
  2. Se le añaden los atributos de las entidades hijas, indicando que pueden quedar en blanco (*)
  3. El atributo “Tipo” se añade a la tabla correspondiente a la entidad padre



– *En este ejemplo vemos que las entidades hijas tienen pocos o ningún atributo, lo que nos puede animar a esta solución*

– *En el ejemplo anterior, con esta solución, un empleado convencional tendría en blanco muchos atributos: licencia, especialidad, empresa, facultad y colegiado*

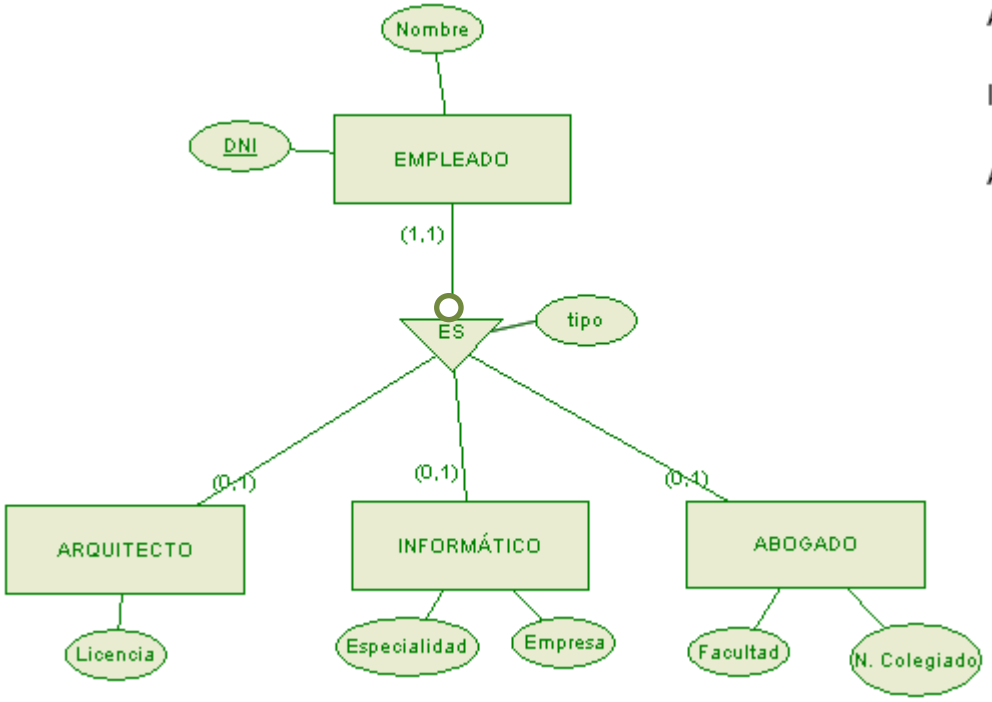
EMPLEADO (DNI, Nombre, Apellido1, Apellido2, Dirección, Licencia*, Especialidad*, **Tipo**)

Paso del modelo E/R al relacional

Jerarquías

- Si vemos que la jerarquía es total y queremos dar más peso a las entidades hijas:

1. Se genera una tabla para cada entidad hija
2. En cada una de ellas se añaden los atributos comunes de la entidad padre
3. La tabla “Empleado” y el atributo “Tipo” ya no son necesarios



ARQUITECTO (DNI, Nombre, Licencia)

INFORMÁTICO (DNI, Nombre, Especialidad, Empresa)

ABOGADO (DNI, Nombre, Facultad, N.Colegiado)

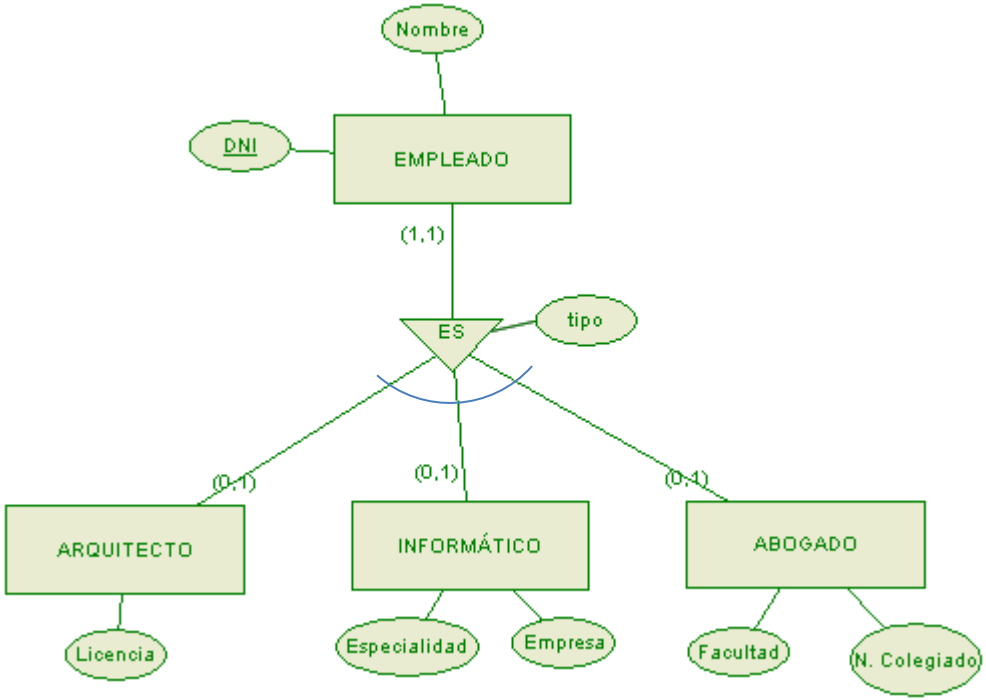
En este ejemplo el protagonismo lo tienen las entidades hijas. La entidad padre apenas aporta atributos comunes.

Al ser total, no habrá Empleados comunes, y eso nos puede convencer de esta solución (aunque podríamos haber usado las otras soluciones igualmente)

Paso del modelo E/R al relacional

Jerarquías

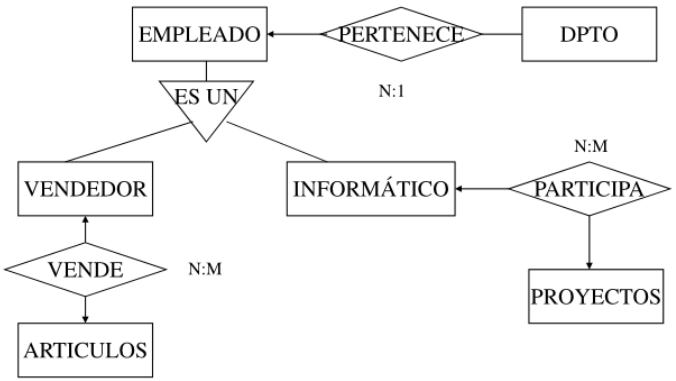
- ¿Cómo se pasan al modelo relacional las restricciones de jerarquías exclusivas/inclusivas, parciales/totales?
  - Más adelante estudiaremos mecanismos para hacer las comprobaciones (asertos, restricciones, comprobaciones y disparadores)



Paso del modelo E/R al relacional

Jerarquías

- Otro ejemplo de la solución equilibrada:



EMPLEADO (<u>COD-EMP</u>, NOMBRE, DNI, SUELDO,COD-DPTO)

COD-DPTO es C.A. en DPTO ya que hay una relación entre EMPLEADO y DPTO

DPTO (<u>COD-DPTO</u>, NOMBRE, PRESUPUESTO)

VENDEDOR (<u>COD-VEN</u>, COMISIÓN, CUOTA)

INFORMÁTICO (<u>COD-INF</u>, CATEGORÍA)

En VENDEDOR e INFORMÁTICO se incluyen los atributos específicos de los subtipos. Como los vendedores e informáticos son empleados tanto COD-VEN como COD-INF son claves ajenas en EMPLEADO, así se obtienen todos los demás atributos comunes.

ARTICULOS (<u>COD-ART</u>, STOCK, PVP)

VENDE (<u>COD-VEN</u>, <u>COD-ART</u>, CANTIDAD)

Como hay una relación N:M entre VENDEDOR y ARTICULOS se crea la tabla VENDE con las respectivas claves ajenas

PROYECTOS (<u>COD-PROY</u>, DESC, PRESUPUESTO)

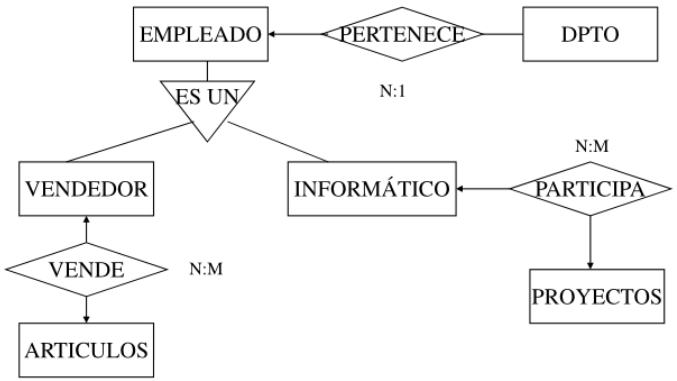
PARTICIPA (<u>COD-INF</u>, <u>COD-PROY</u>, FUNCION)

Idem respecto a la relación PARTICIPA que también es N:M

Paso del modelo E/R al relacional

Jerarquías

- Y si damos prioridad a la entidad padre:



EMPLEADO (<u>COD-EMP</u>, NOMBRE, DNI, SUELDO,COD-DPTO, COMISIÓN, CUOTA, CATEGORÍA)

- *COD-DPTO es C.A. en DPTO ya que hay una relación entre EMPLEADO y DPTO*
- *Los atributos específicos COMISIÓN, CUOTA Y CATEGORÍA se almacenan en EMPLEADO. Para los vendedores el atributo CATEGORÍA quedará a Null y viceversa.*

DPTO (<u>COD-DPTO</u>, NOMBRE, PRESUPUESTO)

ARTICULOS (<u>COD-ART</u>, STOCK, PVP)

VENDE (<u>COD-EMP</u>, <u>COD-ART</u>, CANTIDAD)

- *La relación N:M entre VENDEDOR y ARTICULOS se establece ahora entre EMPLEADO y ARTICULOS. El COD-EMP de VENDE es clave ajena en EMPLEADOS. Idem para PARTICIPA.*

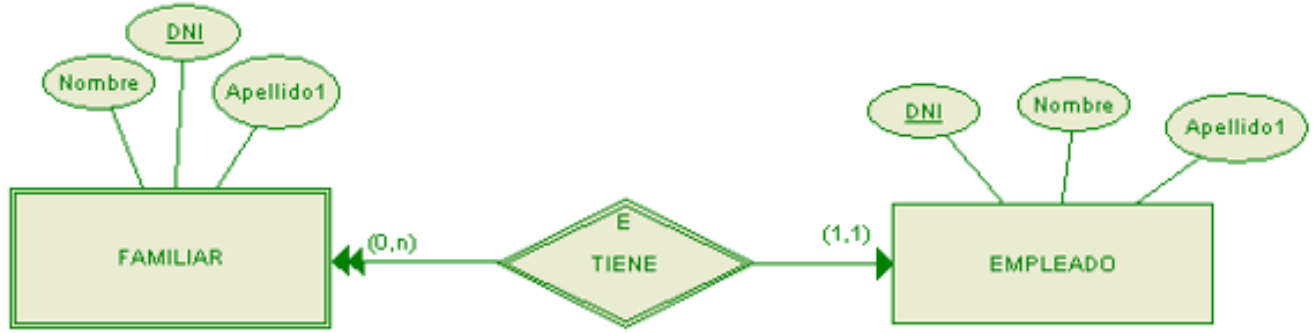
PROYECTOS (<u>COD-PROY</u>, DESC, PRESUPUESTO)

PARTICIPA (<u>COD-EMP</u>, <u>COD-PROY</u>, FUNCION)

Paso del modelo E/R al relacional

Dependencia en existencia

- Se tratan como relaciones 1:n convencionales:
  - La clave se propaga desde la entidad de cardinalidad 1 hacia la de n (que será siempre la débil)
  - Además, habrá que habilitar en nuestro gestor de bases de datos la restricción de **borrado en cascada** para esta relación:
    - Cuando se elimine un empleado de la tabla, todos sus familiares se borrarán también de la base de datos.



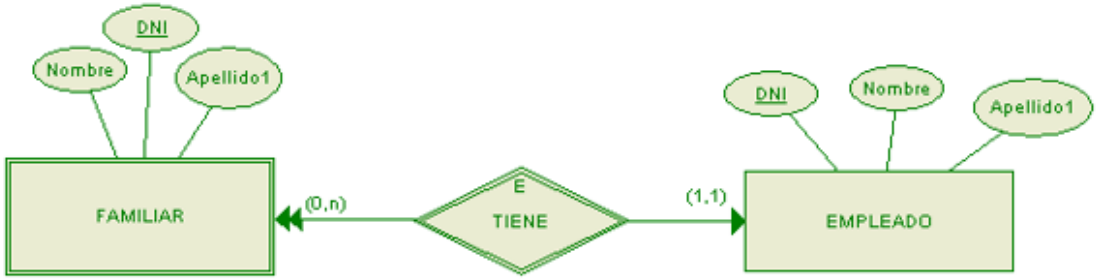
FAMILIAR (DNI, Nombre, Apellido1, DNI_Empleado)

EMPLEADO (DNI, Nombre, Apellido1)

Paso del modelo E/R al relacional

Dependencia en identificación

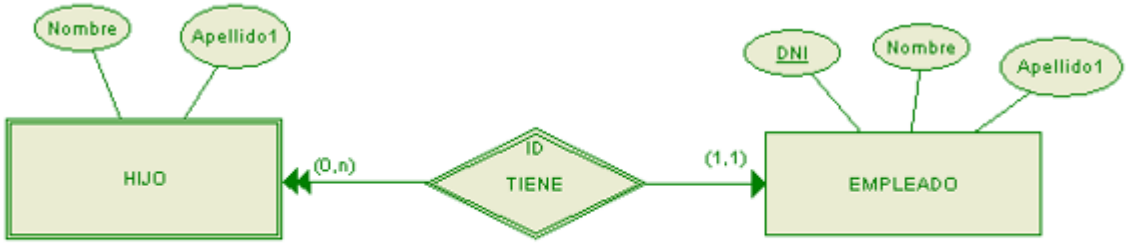
- Se trata igual que la anterior, pero además:
  - La clave propagada hacia la entidad débil formará parte de su clave primaria, ya que esta entidad no puede identificarse por sí misma.



← *En existencia*

FAMILIAR (DNI, Nombre, Apellido1, DNI_Empleado)

EMPLEADO (DNI, Nombre, Apellido1)



← *En identificación*

HIJO (DNI_Empleado, Nombre, Apellido1)

EMPLEADO (DNI, Nombre, Apellido1)

Restricciones

- Las restricciones son reglas que hacen que los datos almacenados en nuestras tablas sean válidos
- A lo largo del tema has visto que hemos ido hablando de restricciones. Es el momento de repasarlas todas, aunque se irán estudiando más adelante:
  - Restricción de clave primaria (Primary Key)
  - Restricción de unicidad (UNIQUE)
  - Restricción de obligatoriedad (NOT NULL)
  - Restricción de clave ajena (Foreign Key), integridad referencial
  - Restricción de verificación (CHECK)
  - Aserciones (ASSERTION)
  - Disparadores (TRIGGER)

Restricciones

- **Clave primaria o principal (Primary Key)**
  - Conjunto mínimo de atributos que identifican de forma unívoca a una tupla
  - Si es uno solo: simple
  - Si tiene más de un atributo: compuesta
- Ejemplo: para la tabla Persona(DNI, NSS, Nombre, Apellidos)
  - Claves candidatas: DNI, NSS
  - Clave primaria: DNI
  - Clave alternativa: NSS
- El/la diseñador/a será quien escoja la clave primaria

Restricciones

- **Unicidad (UNIQUE)**
  - Si un atributo tiene esta restricción, no podrá haber dos valores iguales en la tabla
- **Obligatoriedad (NOT NULL)**
  - Permite definir que los valores de un atributo o conjunto de atributos no tomen valores NULL (vacíos)

- Todos los atributos que son parte de la clave primaria necesariamente son UNIQUE y NOT NULL

Restricciones

- **Integridad referencial (FOREIGN KEY)**
  - El concepto de clave ajena se estudió antes
  - La integridad referencial garantiza que todo valor de la clave ajena se corresponde con un valor de clave primaria en la tabla relacionada

LIBRO (ISBN, Título)

ESCRIBE (ISBN Libro, NIF_Autor)

AUTOR(NIF, Nombre, Nacimiento)

Escribe

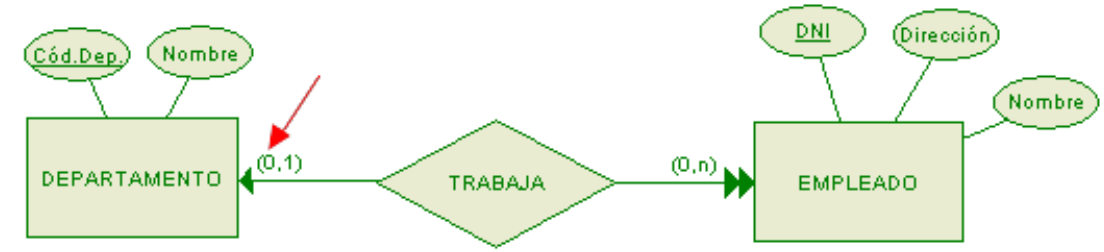
ISBN_Libro	NIF_Autor
345345345345	555234K
11232334432	444656L

La restricción de integridad referencial garantiza que:

- *En la tabla AUTOR habrá algún autor con NIF 555234K*
- *En la tabla LIBRO habrá algún libro con el ISBN 345345345*

Restricciones

- **Integridad referencial (FOREIGN KEY)**
  - Las claves ajenas NO siempre son NOT NULL
  - Recuerda este caso: es posible que el empleado no pertenezca a ningún departamento



DEPARTAMENTO (Cód.Dep, Nombre)

EMPLEADO (DNI, Dirección, Nombre, Cód.Dep *)

Restricciones

- **Integridad referencial (FOREIGN KEY)**
- Ante el borrado o modificación de las tablas AUTOR o LIBRO, ¿qué sucede con la tabla ESCRIBE?
- Hay que indicar el **Modo de Borrado y Modificación:**
  - **Operación restringida (NO ACTION/RESTRICTED)**
    - Si la operación rompe la restricción referencial, no se permite.
  - **Operación con transmisión en cascada (CASCADE)**
    - Propagar la modificación (o borrado) de las tuplas de la tabla que referencia.
      - » Lo vimos para entidades débiles
  - **Operación con puesta a nulo (SET NULL).**
    - Los valores afectados serán sustituidos por el valor nulo.
  - **Operación con puesta a valor por defecto (SET DEFAULT)**
    - Los valores afectados serán sustituidos por un valor por defecto.

Restricciones

- **Restricciones de comprobación de atributos de una relación (CHECK)**
  - Sólo hará operaciones si se cumple una condición de atributos de la tabla:
    - "El saldo debe ser mayor que 0"
    - "El NIF serán 8 dígitos y una letra"
    - "El nombre se introducirá en mayúsculas"

- **Restricciones entre atributos de varias relaciones (ASSERTION)**
  - Igual, pero afecta a varias tablas, más compleja
    - "El saldo en esta tabla será mayor que el importe del préstamo en la tabla "Préstamo"

Restricciones

- **Disparadores (TRIGGER)**
  - Permiten al diseñador especificar cómo debe reaccionar el sistema cuando se satisface una condición, mediante lenguajes de programación
    - “Cuando se cree una nueva venta, añadir una fila en la tabla “AUDITORÍA” indicando cuándo ha sucedido

En resumen

- Entidades: nueva tabla
- Relaciones:
  - N:N → Nueva tabla
  - 1:N → Nuevo atributo en la entidad de la N.
    - Decidir si es opcional, según si el lado del 1 tiene cardinalidad mínima 0 o 1
  - 1:1 → Caso particular de 1:N, nuevo atributo también en la que creas más conveniente
  - Grado 1 → Nada diferente. Imaginar que son dos entidades diferentes
  - Grado 3 o más → Nueva tabla. La clave primaria es la unión de las claves de todas las entidades participantes
- Atributos multivaluados: nueva tabla
- Jerarquías:
  - Solución equilibrada → tablas para entidades padre e hijas
  - Más peso para padre → una única tabla para la entidad padre con los atributos opcionales que provengan de las hijas
  - Más peso para las hijas → en jerarquías totales se puede quitar la tabla de la entidad padre y usar una tabla para cada hija
- Dependencias en existencia e identificación: se aplica borrado en cascada, pero no se representa en el esquema