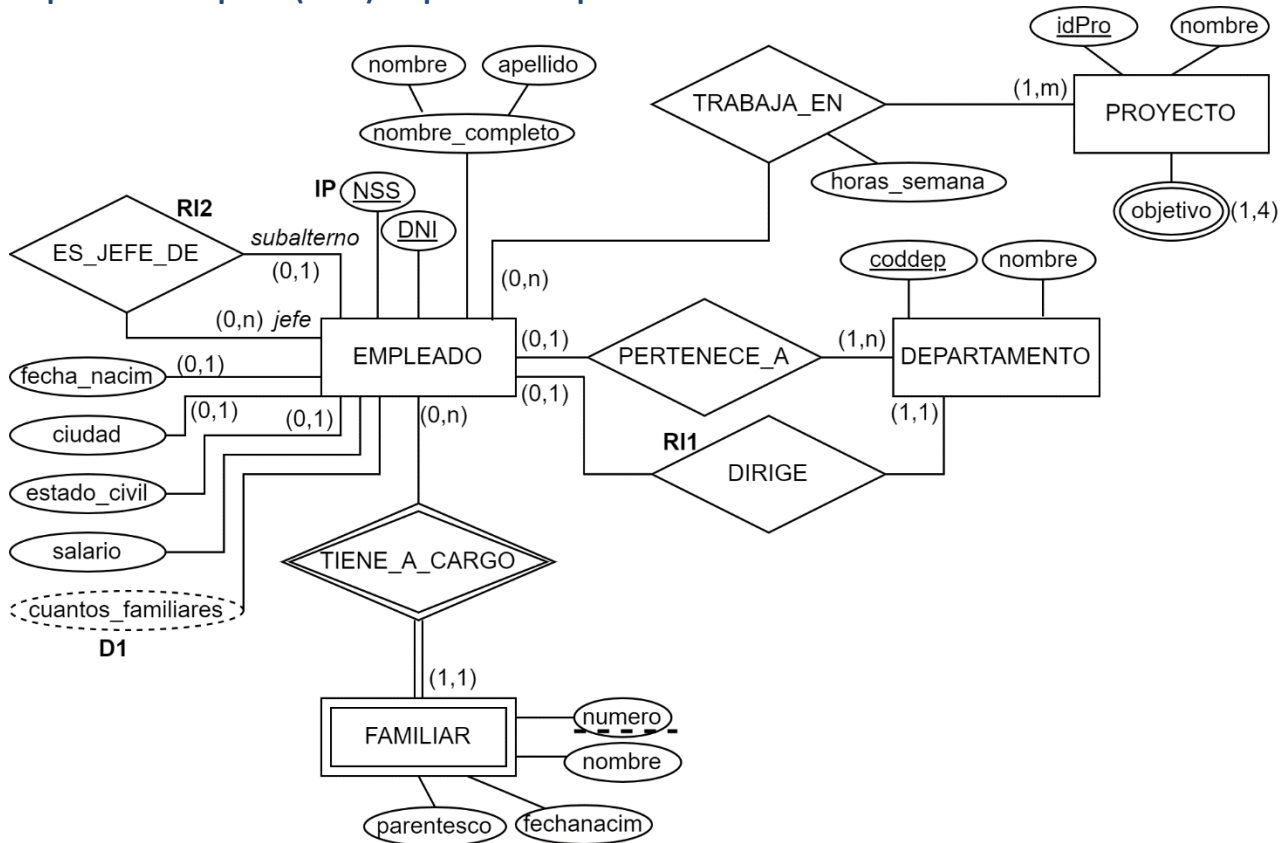


El ejercicio consiste en **traducir a relaciones (Modelo Relacional) los elementos que integran el esquema conceptual de datos** mostrado a continuación.

Para describir cada relación mediante una ficha, se debe utilizar la plantilla proporcionada por los profesores de la asignatura (y empleada en el tema de teoría).

Esquema Conceptual (MER) de partida. Empresa.



Restricciones de integridad de Dominio

- EMPLEADO.estado_civil IN ('S', 'C', 'V', 'D', 'P')
- EMPLEADO.salario > 0
- FAMILIAR.numero > 0
- TRABAJA_EN.horas_semana > 0

Fórmulas de cálculo de Atributos Derivados

D1: EMPLEADO.cuantos_familiares = contar instancias de FAMILIAR conectadas con cada instancia de EMPLEADO vía TIENE_A_CARGO.

Restricciones de integridad Generales (Asertos)

RI1. Un empleado sólo puede estar conectado vía DIRIGE al departamento al que está vinculado vía PERTENECE_A
RI2. Un empleado no puede ser jefe de sí mismo: una instancia de EMPLEADO que participa en ES_JEFE_DE no puede estar conectada a ella misma.

Solución:

Diseño Lógico Estándar: traducción a relaciones del Modelo Relacional Formal.

Elemento MERE	Elemento MR
Tipos de Entidad	Relaciones
EMPLEADO	EMPLEADO
DEPARTAMENTO	DEPARTAMENTO
FAMILIAR	FAMILIAR
PROYECTO	PROYECTO
Tipos de Relación	
PERTENECE_A	Clave ajena en EMPLEADO que referencia a DEPARTAMENTO; admite nulos
DIRIGE	Clave ajena en DEPARTAMENTO que referencia a EMPLEADO; no admite nulos
TIENE_A_CARGO	Clave ajena en FAMILIAR que referencia a EMPLEADO; no admite nulos, parte de la PK
ES_JEFE_DE	Clave ajena en EMPLEADO que referencia a EMPLEADO; admite nulos
TRABAJA_EN	Relación EMPLEADO_EN_PROYECTO, con dos claves ajenas: una referencia a EMPLEADO y otra referencia a PROYECTO
Atributos Multivalorados	
objetivo (en PROYECTO)	Relación OBJETIVO_PROYECTO, con una clave ajena que referencia a PROYECTO

Fichas de descripción de Relaciones obtenidas en la traducción:

EMPLEADO (nss, dni, nombre, apellido, fechanacim, ciudad, est_civil, salario, cuantos_familiares, nssjefe, dep)

Admiten NULL: fechanacim, ciudad, estado_civil, dep, nssjefe

Clave primaria: nss

Claves alternativas (UNIQUE):

1. dni	2. --	3.--
--------	-------	------

Claves ajenas (FOREIGN KEY):

1. dep	Referencia_a	DEPARTAMENTO(coddep)
2. nssjefe	Referencia_a	EMPLEADO(nss)
3.	Referencia_a	

Derivados:

cuantos_familiares: para cada fila e, contar el número de filas f de FAMILIAR cuyo f.nssemp = e.nss

...

Comprobar:

salario > 0

est_civil IN ('S', 'C', 'V', 'D', 'P')

nss <> nssjefe

DEPARTAMENTO (coddep, nombre, nssdire)

Admiten NULL: ninguno

Clave primaria: **coddep**

Claves alternativas (UNIQUE):

1. nssdire	2. --	3.--
-------------------	-------	------

Claves ajenas (FOREIGN KEY):

1. nssdire	Referencia_a	EMPLEADO(nss)
2.	Referencia_a	
3.	Referencia_a	

Derivados:

...

Comprobar:

FAMILIAR (numero, nombre, parentesco, fechanacim, nssemp)

Admiten NULL: ninguno

Clave primaria: **(nssemp, numero)**

Claves alternativas (UNIQUE):

1. --	2. --	3.--
-------	-------	------

Claves ajenas (FOREIGN KEY):

1. nssemp	Referencia_a	EMPLEADO(nss)
2.	Referencia_a	
3.	Referencia_a	

Derivados:

...

Comprobar:

numero > 0

PROYECTO (idPro, nombre)

Admiten NULL: ninguno

Clave primaria: **idPro**

Claves alternativas (UNIQUE):

1. --	2. --	3.--
-------	-------	------

Claves ajenas (FOREIGN KEY):

1.	Referencia_a	
2.	Referencia_a	
3.	Referencia_a	

Derivados:

...

Comprobar:

EMPLEADO_EN_PROYECTO (empleado, proyecto, horas_semana) – proviene del tipo de relación TRABAJA_EN

Admiten NULL: ninguno

Clave primaria: **(empleado, proyecto)**

Claves alternativas (UNIQUE):

1. --	2. --	3.--
-------	-------	------

Claves ajenas (FOREIGN KEY):

1.	empleado	Referencia_a	EMPLEADO(nss)
2.	proyecto	Referencia_a	PROYECTO(idPro)
3.		Referencia_a	

Derivados:

...

Comprobar:

horas_semana > 0

OBJETIVO_PROYECTO (proyecto, objetivo) – proviene del atributo multivalorado PROYECTO.objetivo

Admiten NULL: ninguno

Clave primaria: **(proyecto, objetivo)**

Claves alternativas (UNIQUE):

1. --	2. --	3.--
-------	-------	------

Claves ajenas (FOREIGN KEY):

1.	proyecto	Referencia_a	PROYECTO(idPro)
2.		Referencia_a	
3.		Referencia_a	

Derivados:

...

Comprobar:

(sigue→)

ASERTOS surgidos durante la traducción

1. Todo departamento ha de tener al menos un empleado.

ASERTO A1.

```
COMPROBAR_QUE (  
  NO_EXISTE (una tupla de DEPARTAMENTO  
    tal que el valor de "coddep"  
      NO_ESTÉ_ENTRE (valores de "dep"  
        de EMPLEADO)));
```

2. Todo proyecto ha de tener al menos un empleado asignado.

ASERTO A2.

```
COMPROBAR_QUE (  
  NO_EXISTE (una tupla de PROYECTO  
    tal que el valor de "idPro"  
      NO_ESTÉ_ENTRE (valores de "proyecto"  
        de EMPLEADO_EN_PROYECTO)));
```

3. Todo proyecto ha de tener al menos un objetivo.

ASERTO A3.

```
COMPROBAR_QUE (  
  NO_EXISTE (una tupla de PROYECTO  
    tal que el valor de "idPro"  
      NO_ESTÉ_ENTRE (valores de "proyecto"  
        de OBJETIVO_PROYECTO)));
```

4. Todo proyecto ha de tener un máximo de 4 objetivos.

ASERTO A4.

```
COMPROBAR_QUE (  
  NO_EXISTE (una tupla t de PROYECTO  
    tal que 4 < (CUENTA de las tuplas de OBJETIVO_PROYECTO  
      donde proyecto = t.idPro) ));
```