

Facultad de Ingeniería



Diseño lógico de una base de datos

Tema IV

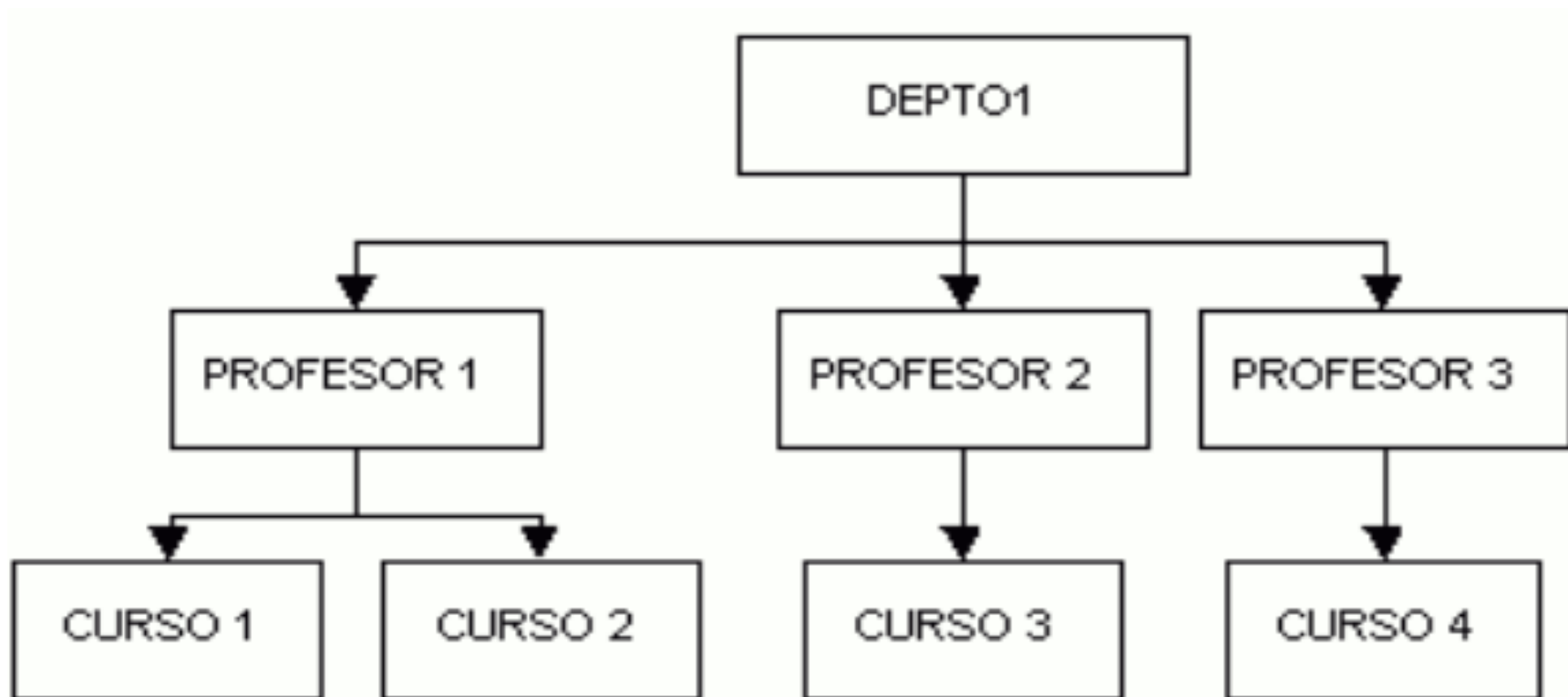
Semestre 2025-2



UNAM
UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO

El alumno realizará la construcción de modelos relacionales a partir de modelos entidad/relación, haciendo uso de conceptos, principios y buenas prácticas, para obtener el diseño lógico de la base de datos. Comprenderá el uso de herramientas CASE empleando diversas notaciones.

Modelo basado en lógica de predicados y en teoría de conjuntos, propuesto en los años 70's por Frank Codd



Relaciones padre-hijo

Propiedades:

- **No pueden existir dos relaciones que se llamen igual**
- **No pueden existir tuplas iguales**
- **No pueden existir atributos que tengan el mismo nombre**
- **No hay orden en tuplas ni en atributos**
- **Los valores de los atributos deben ser atómicos**

Notación Crow's foot

nombre_Relacion



Notación Crow's foot



Transformación de entidades fuertes

- **Toda entidad fuerte se transforma en una relación**
- **Se conservan los atributos y la clave principal (ahora se llamará *llave primaria*, denotada por PK)**

Transformación de atributos

- **En claves candidatas debe establecerse restricción de unicidad (U)**
- **Los atributos compuestos deben indicarse de forma individual**

Transformación de atributos

- **Para atributos multivalor se crea una nueva relación y se propaga como *llave foránea* (FK) la PK de la relación base a la nueva relación**

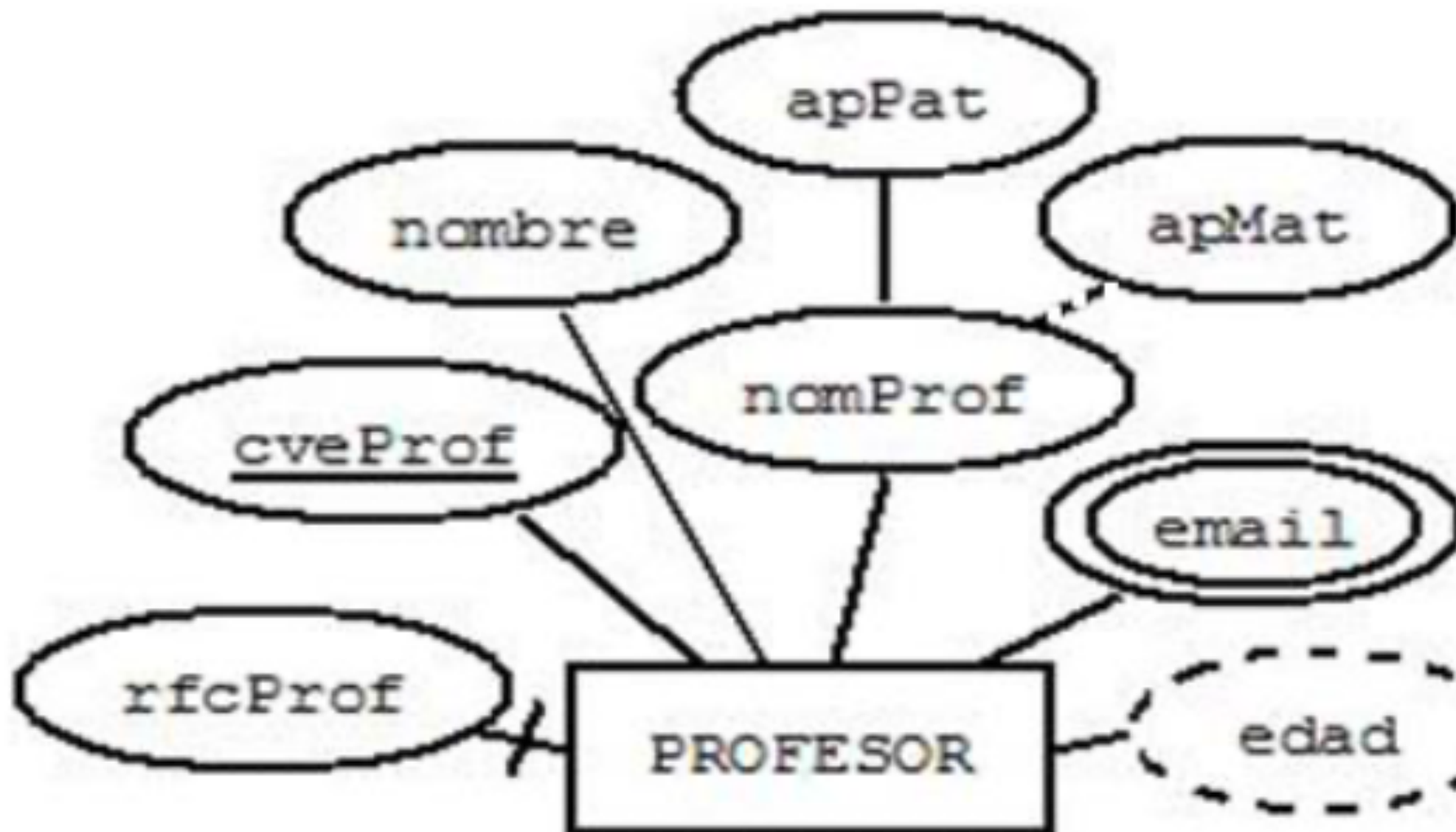
Transformación de atributos

- **Para atributos derivados se indica que son calculados (C)**

Transformación de atributos

- **Finalmente, se deben indicar las restricciones que haya sobre los atributos, como *check* (CK) o *null* (N)**

Ejemplo



***PROFESOR: { cveProf int (PK),
rfcProf varchar(13) (U),
nombre varchar(70),
apPat varchar (50),
apMat varchar (50) (N),
edad smallint (C) }***

***EMAIL: { email varchar(150) (PK),
cveProf int (FK) }***

Transformación de relaciones

- **m:m -> Se crea una nueva relación, que tendrá como PK las PK's de las entidades que une (que a su vez son FK's), más los atributos (si hubiera) de la relación**

Transformación de relaciones

- **1:m ó m:1 -> La llave primaria de la relación con cardinalidad 1 se propaga como llave foránea a la relación con cardinalidad m**

Transformación de relaciones

- **1:1 -> La clave primaria de una relación se propaga a la otra relación dependiendo de:**
 - 1) La semántica**
 - 2) Considerar cuál relación será accedida más frecuentemente**

Transformación de relaciones recursivas

El mapeo se realiza en función de su cardinalidad; se debe tener en consideración que en una relación no pueden existir atributos que se llamen igual

Transformación de entidades débiles

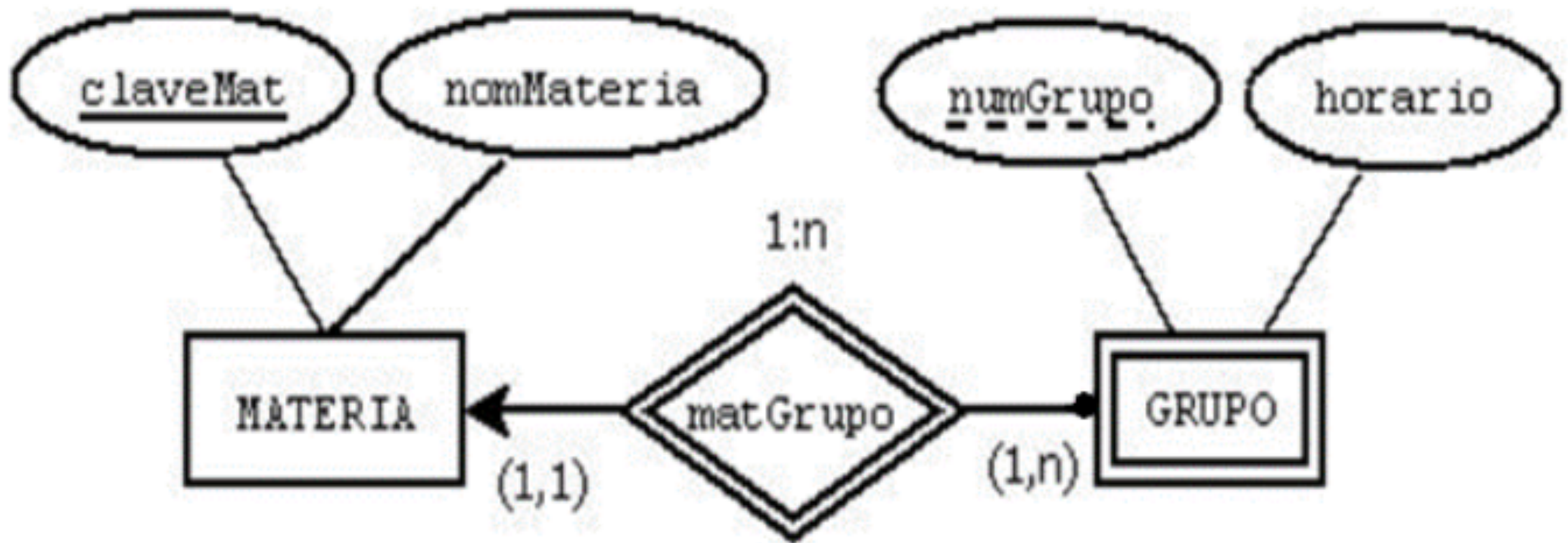
- **Se crea una nueva relación conservando todos sus atributos.**

Transformación de entidades débiles

Se tiene dep. de identificación:

- **Se propaga la llave principal de la entidad fuerte hacia la débil, ya que en conjunto con el discriminante, formará la llave primaria (PK) de la entidad débil.**

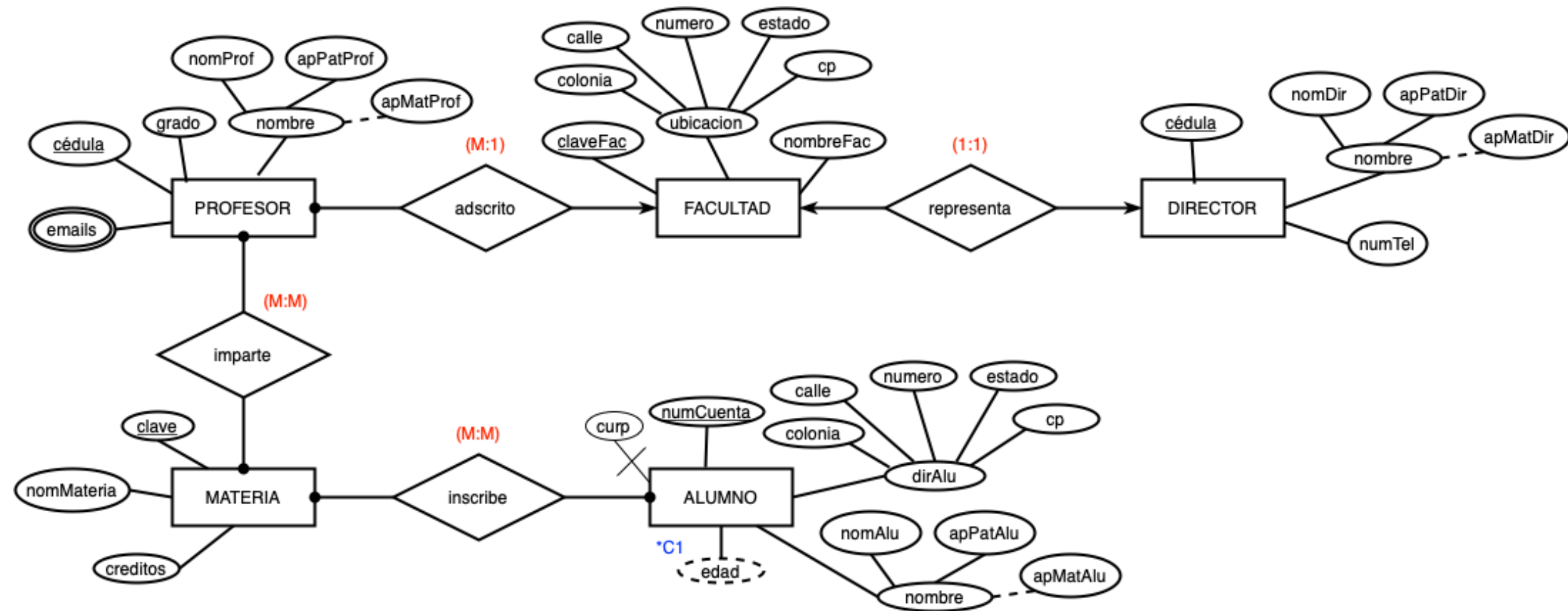
Ejemplo



***MATERIA: { claveMat smallint (PK),
nomMateria varchar (60) }***

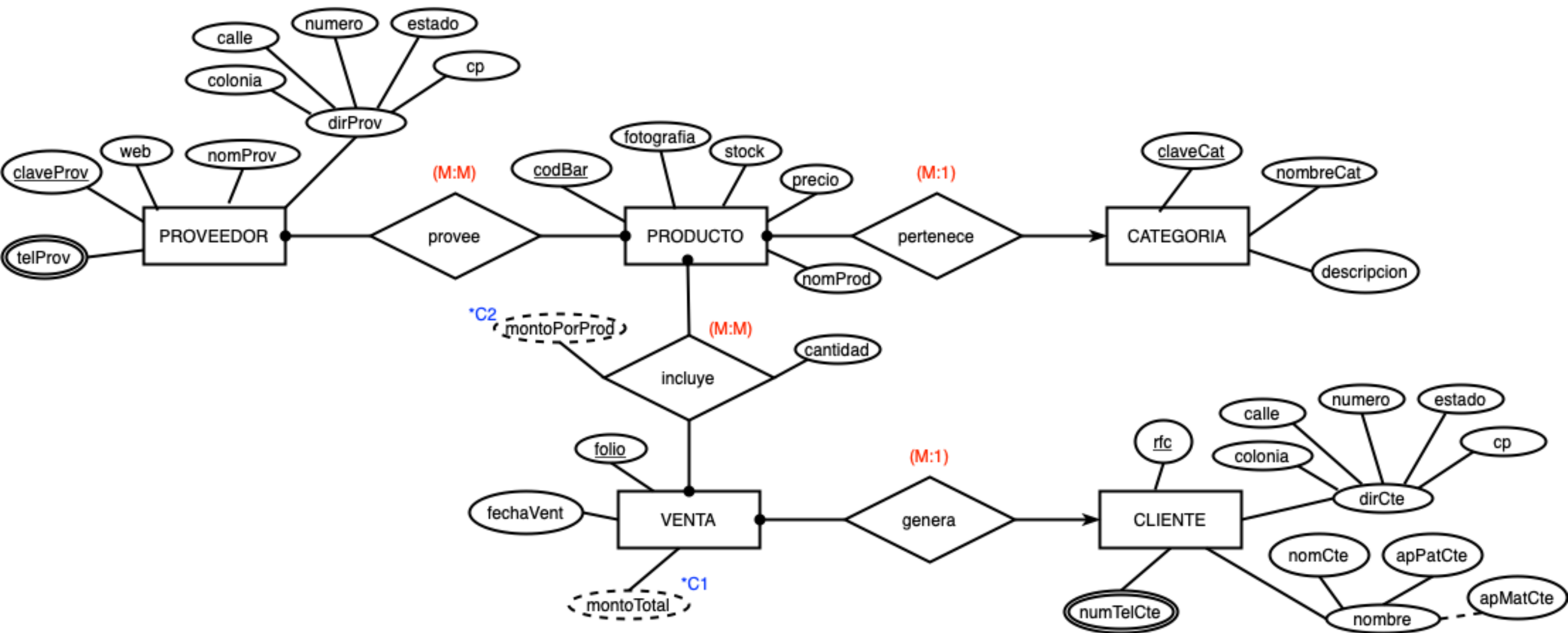
***GRUPO: { [numGrupo smallint (D),
claveMat smallint (FK)] (PK),
horario date}***

Ejercicio 1_4



C1: Calculado a partir del atributo curp de la entidad alumno

Ejercicio 2_4



C1: Calculado a partir del atributo montoPorProd de la relación incluye para cada orden
C2: Calculado a partir del atributo precio de la entidad producto multiplicado por la cantidad de producto que se está incluyendo en la orden

**Busca generar una representación
del modelo conceptual con un
nivel de abstracción mayor.**

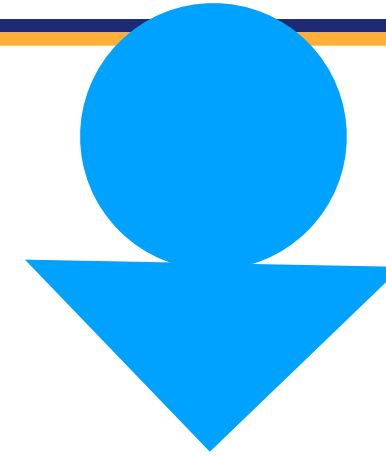
Supertipo: Entidad que es definida por un conjunto de dos o más entidades.

Subtipo: Una entidad que en conjunto con otras, forma parte de un supertipo.

**Los subtipos heredan los atributos
de los supertipos.**

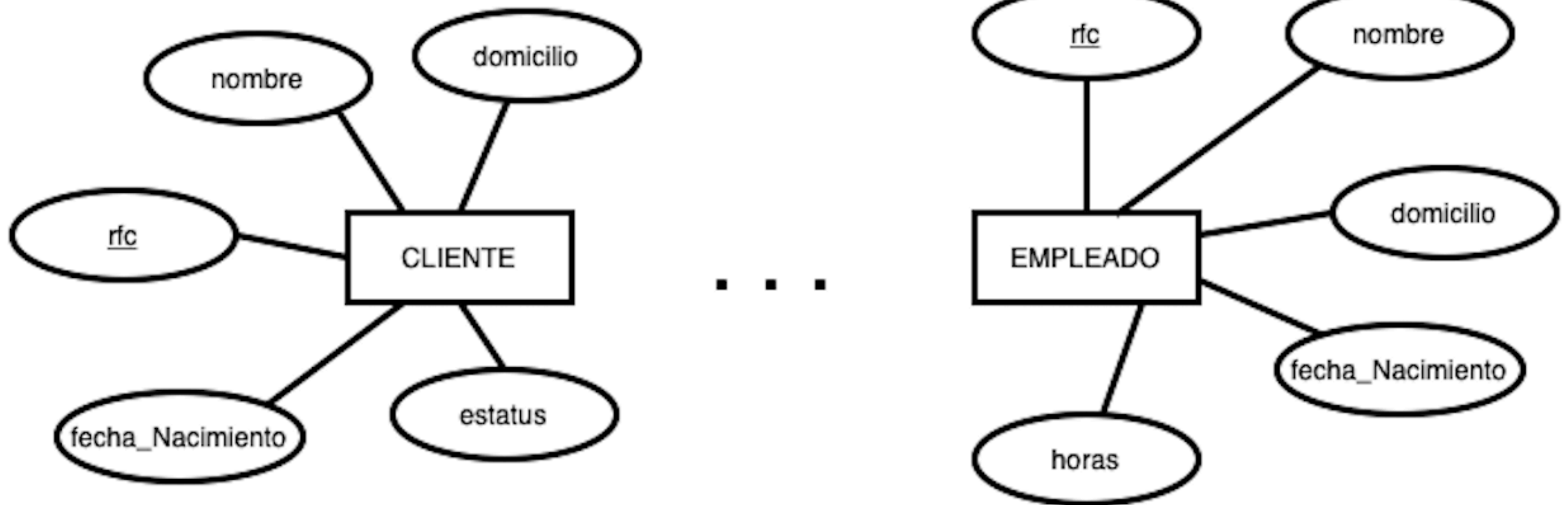
¿Cuándo emplear MER extendido?

Generalización:

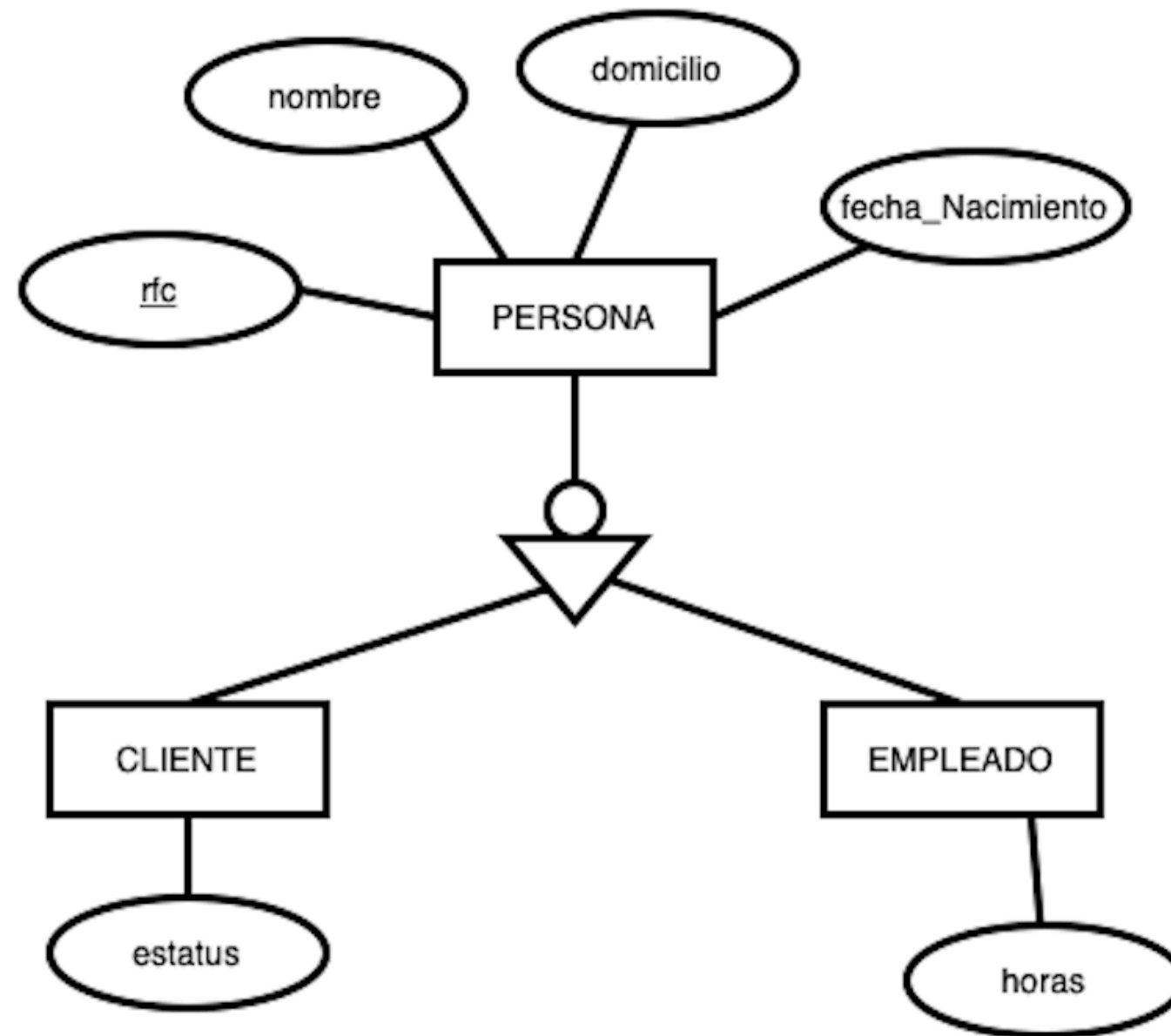


Partiendo de entidades que tienen atributos en común, se puede realizar un proceso de generalización creando una entidad de nivel superior.

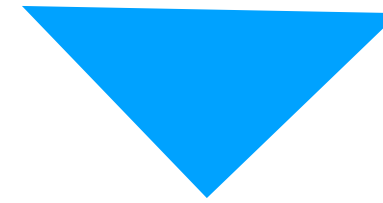
Generalización



Generalización

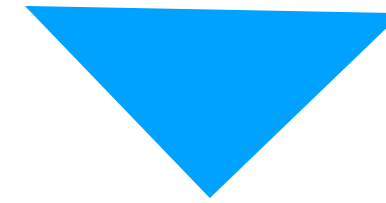


Especialización:



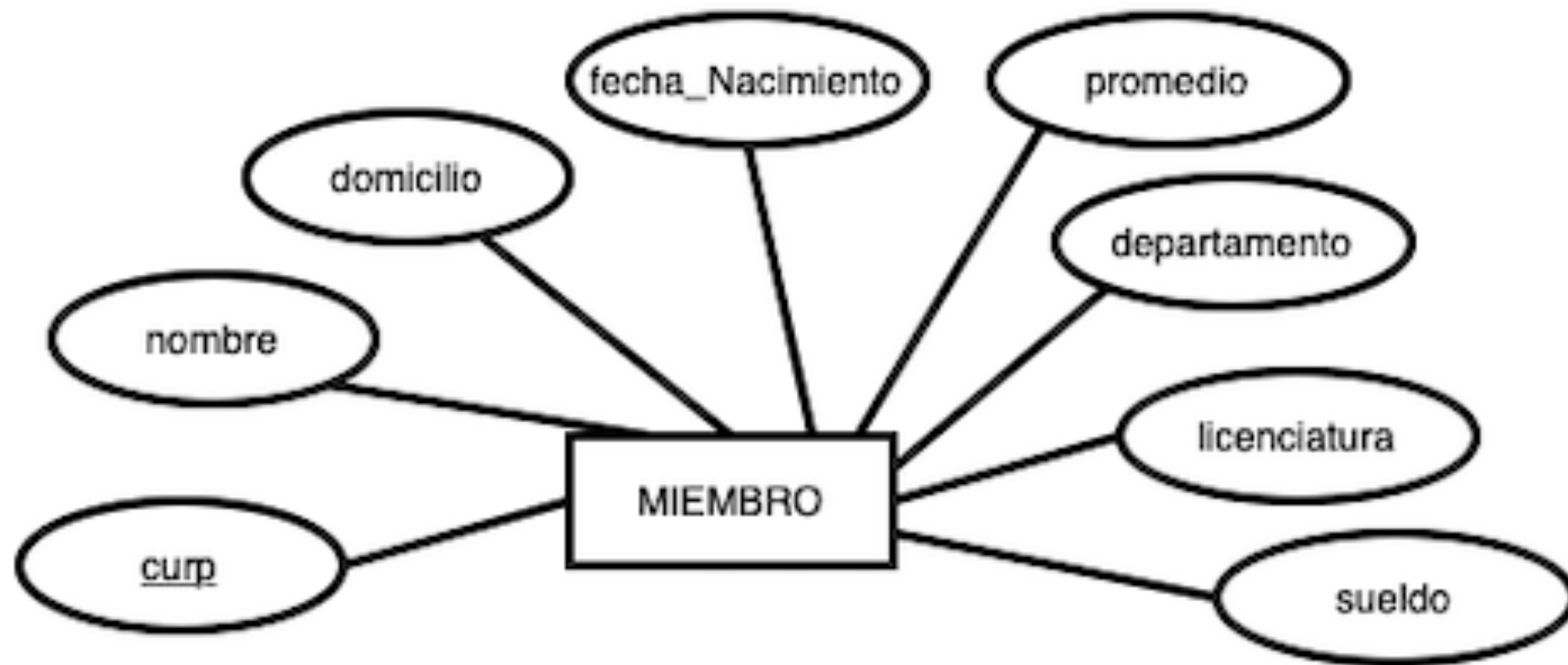
Partiendo de una entidad que tiene ciertos atributos que tienen sentido para parte de los miembros de la entidad, pero para otros no.

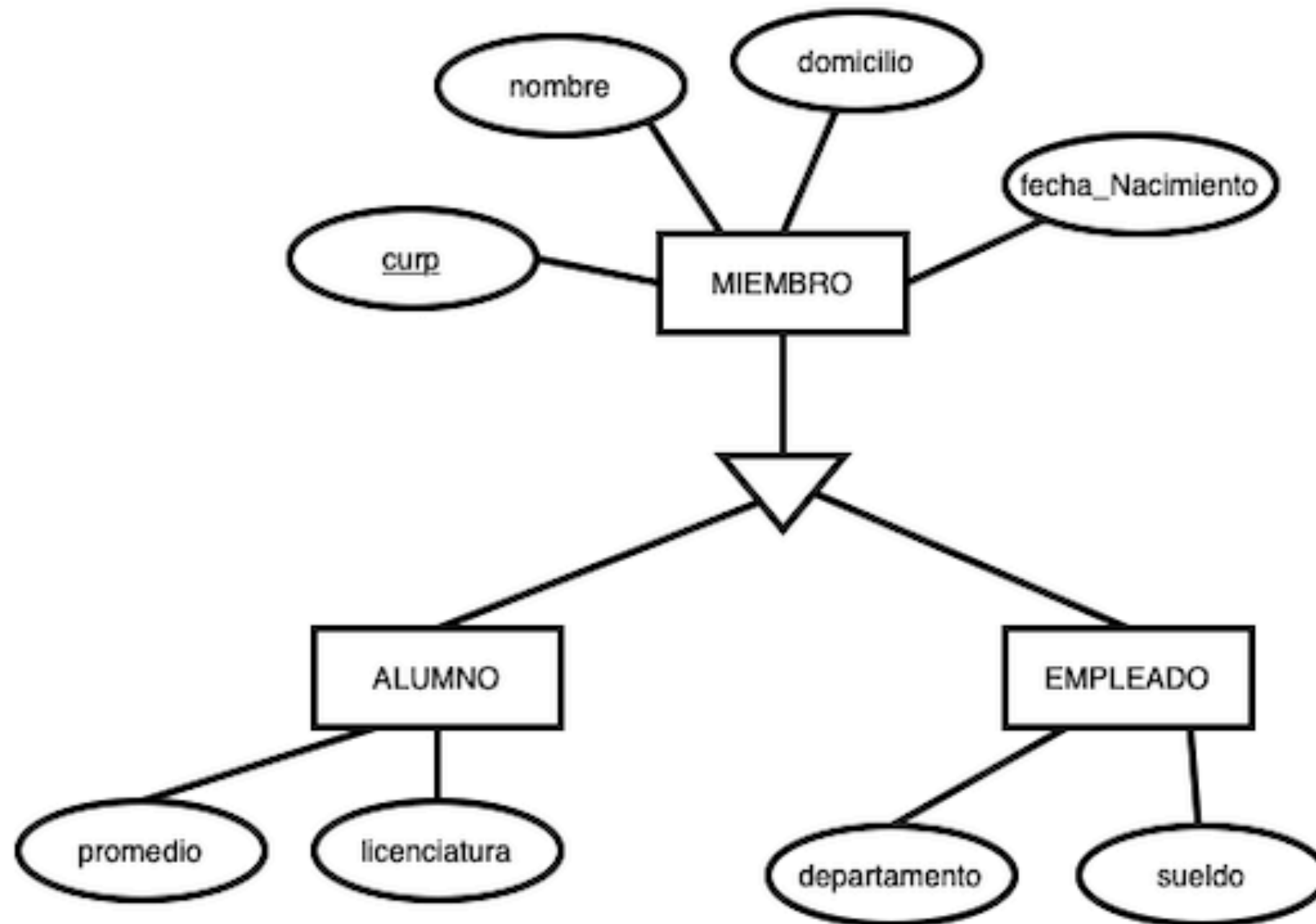
Especialización:



Procedemos a definir subtipos que tengan atributos particulares, manteniendo los comunes en el supertipo.

Especialización





MER extendido

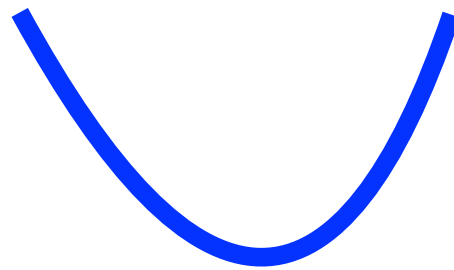


Generalización	Especialización
Enfoque bottom-up	Enfoque top-down
Múltiples entidades convergen en una	Una entidad se separa en múltiples entidades
Reduce el tamaño de la solución *	Aumenta el tamaño de la solución *

Exclusión - traslape

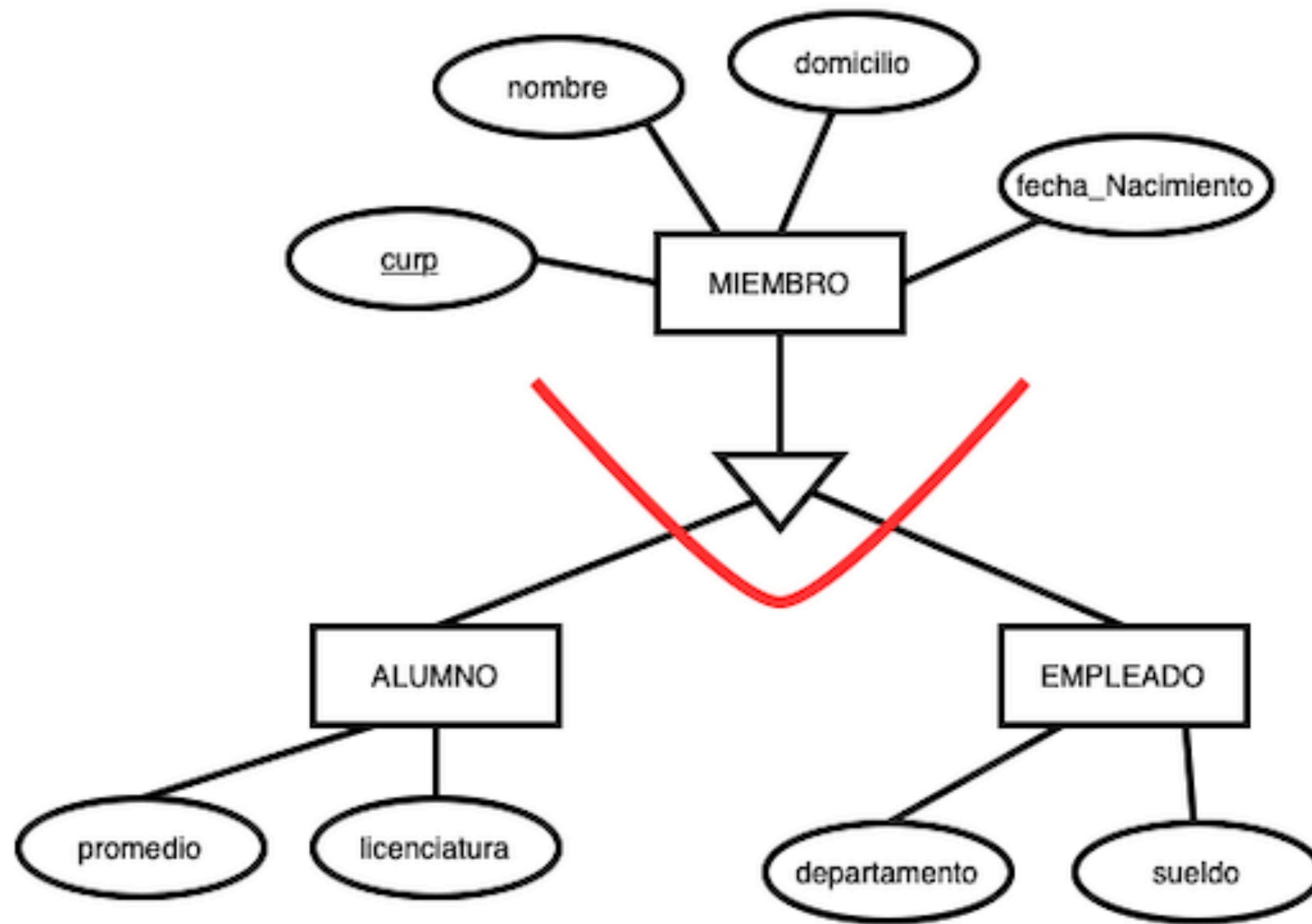
Exclusión: Un supertipo no puede pertenecer a más de un subtipo.

(d)



Traslape: Un supertipo puede pertenecer a más de un subtipo (o)

Restricciones

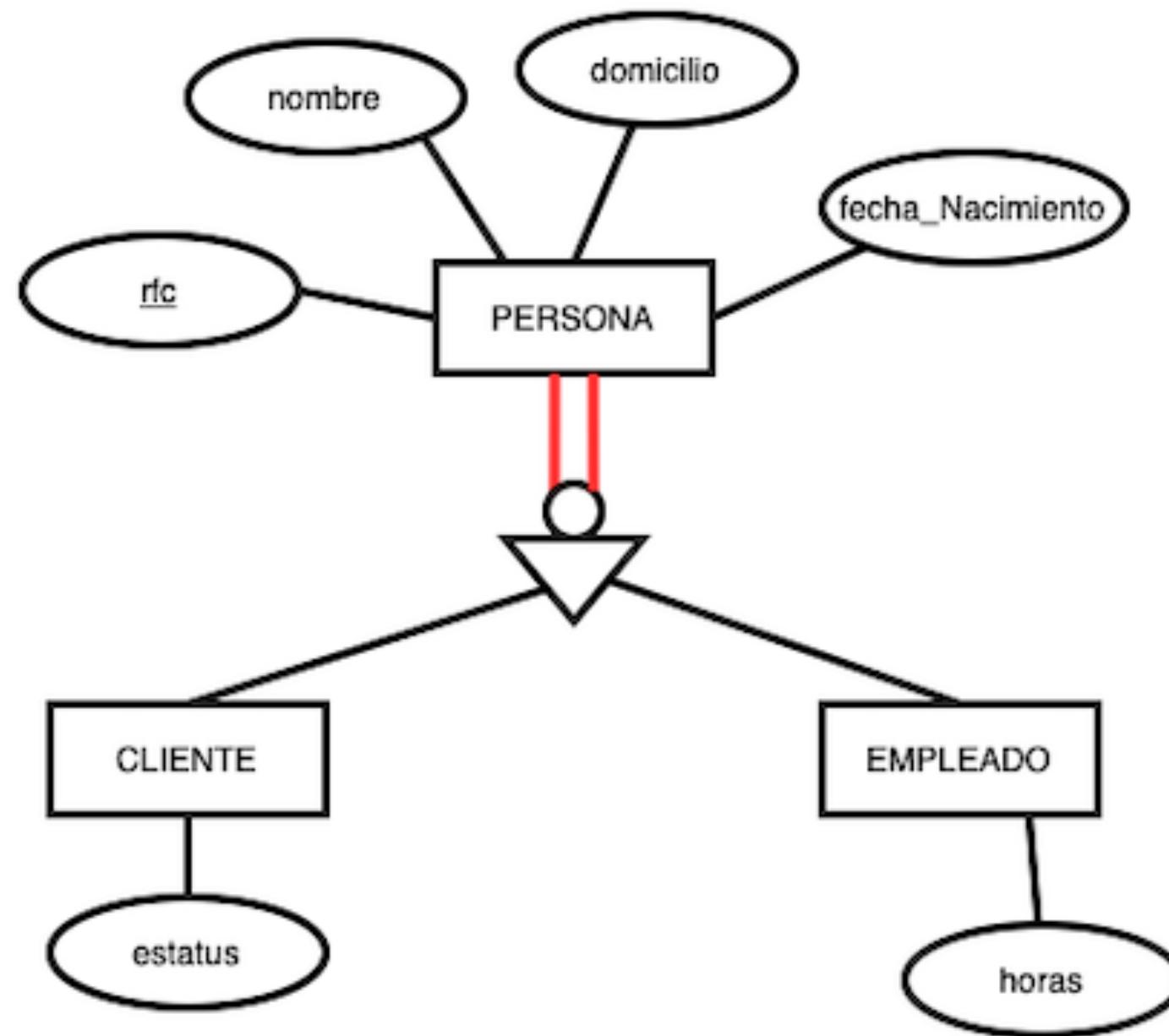


Relación parcial - total **“completenses constraint”**

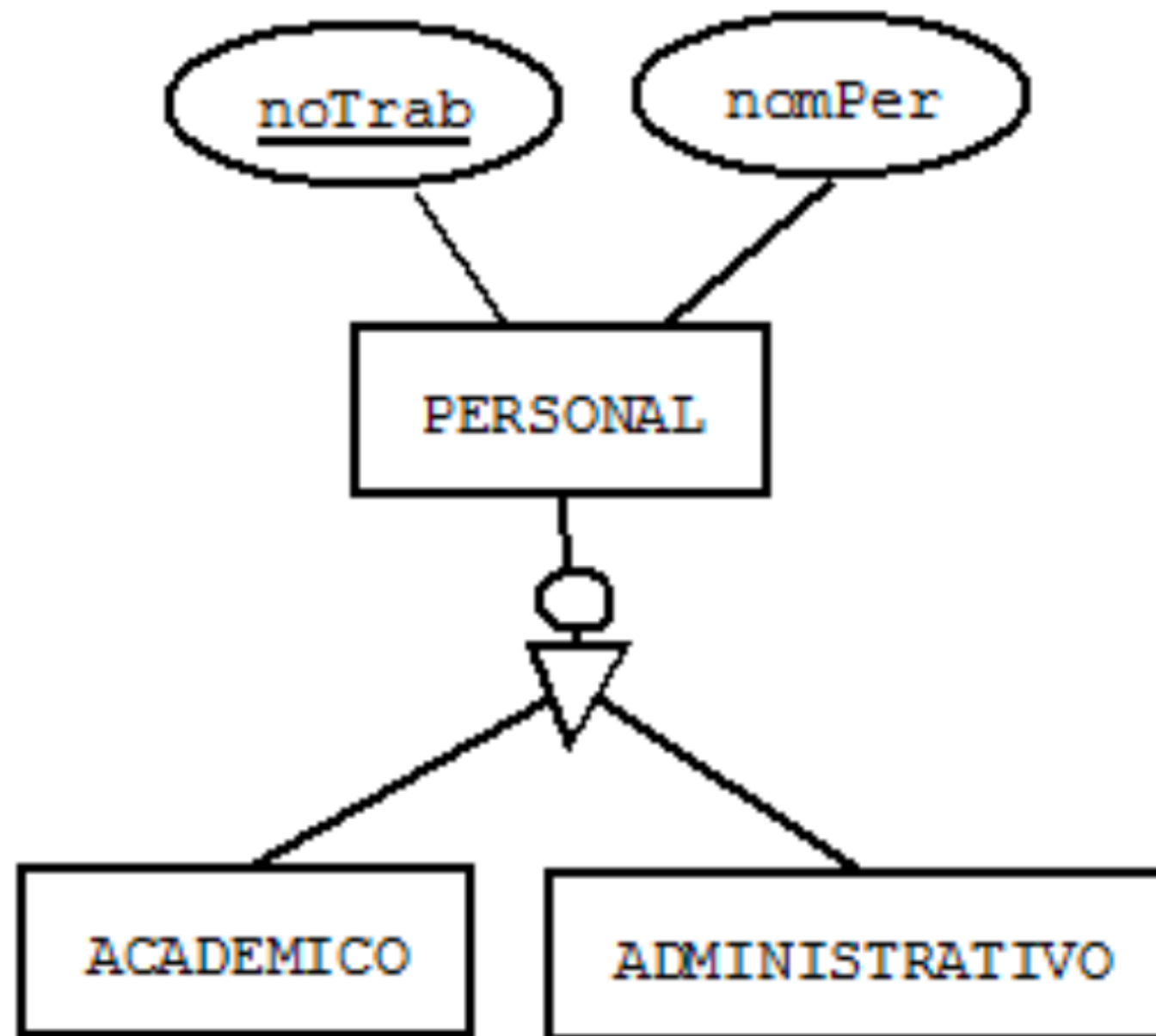
Total: Un supertipo debe pertenecer al menos a un subtipo.

Parcial: Un supertipo puede o no pertenecer a un subtipo.

Restricciones



Restricciones



Una universidad desea construir una BD en donde se dará seguimiento a 3 tipos de personas y tomando en cuenta los siguientes aspectos:

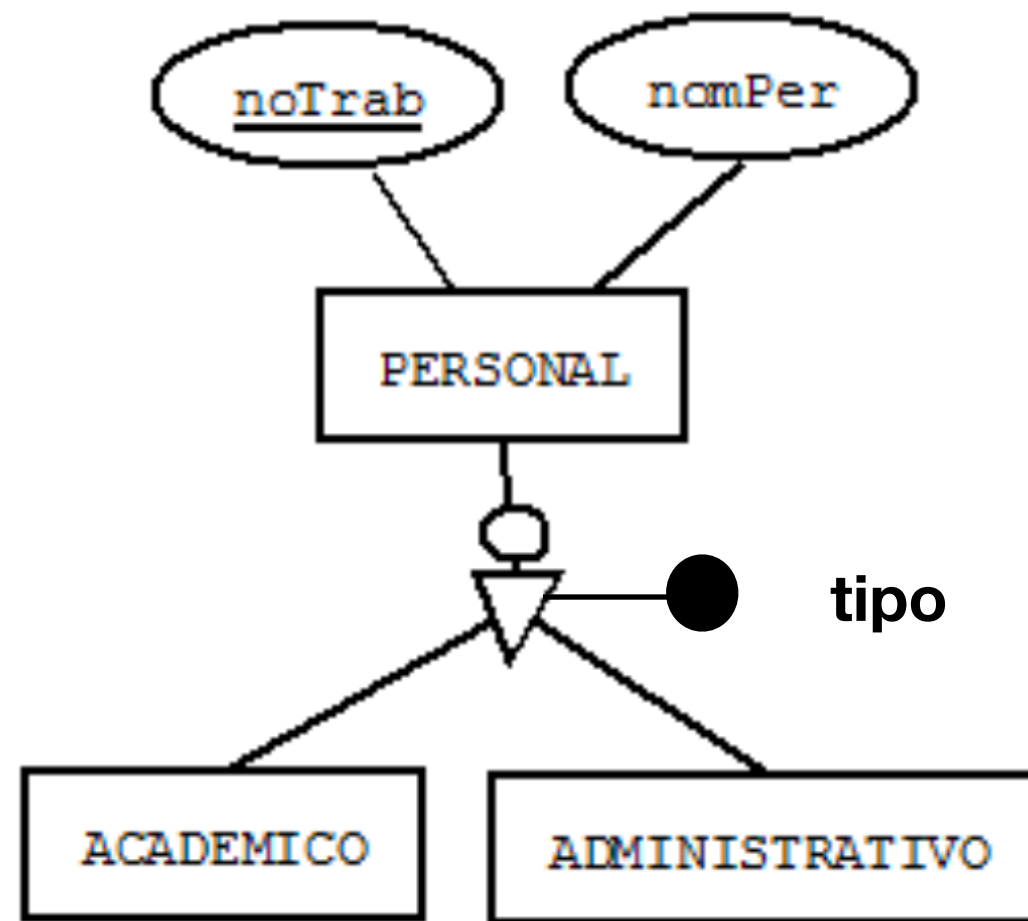
Se dará seguimiento a los alumnos, egresados y a los empleados. Cada persona tiene un nombre, un nss, sexo, dirección y fecha de nacimiento, y debe tenerse en cuenta que una persona puede tener más de un rol. Un empleado tiene asignado un salario y hay distintos tipos: personal en general, profesores y ayudantes, un empleado puede pertenecer sólo a una categoría. De los egresados se debe tener registro de los grados/cursos que obtuvieron en la universidad, almacenando el nombre y fecha de obtención.

De los profesores se tiene su tipo, mientras que del personal en general se debe guardar su puesto. En el caso de los ayudantes, pueden ser de investigador o de profesor, almacenando el proyecto o materia en que laboran, respectivamente, así como sus horas asignadas. De los estudiantes debe tenerse almacenado el depto. al que pertenecen, así como distinguir si son de posgrado(almacenar el programa en que están) o licenciatura, donde debe guardarse su avance de créditos.



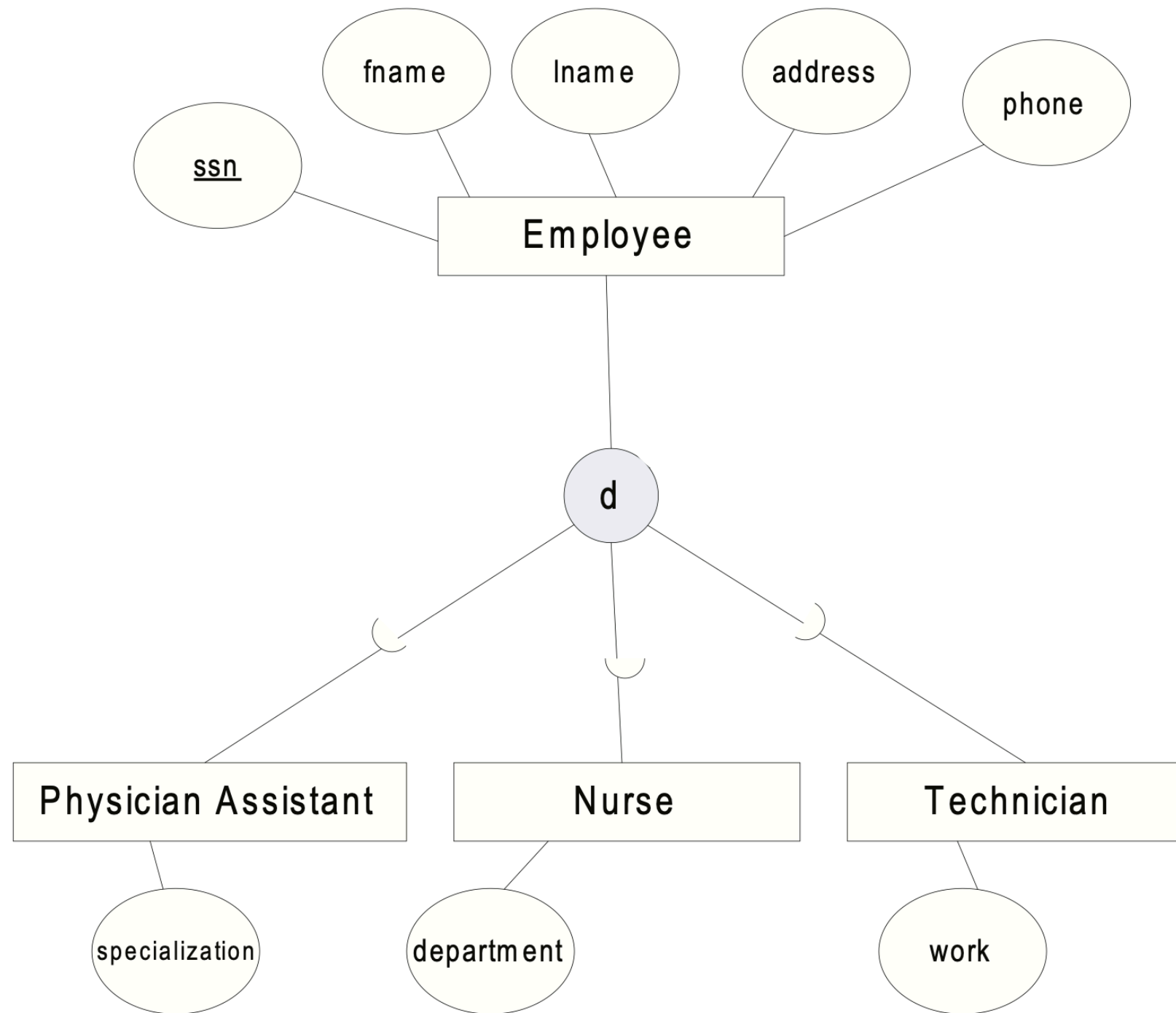
Se dará seguimiento a los **alumnos, egresados y a los empleados**. Cada **persona** tiene un nombre, un nss, sexo, dirección y fecha de nacimiento, y debe tenerse en cuenta **que una persona puede tener más de un rol**. Un empleado tiene asignado un salario y hay distintos tipos: **personal en general, profesores y ayudantes**, un empleado **puede pertenecer sólo a una categoría**. De los egresados se debe tener registro de los grados/cursos que obtuvieron en la universidad, almacenando el nombre y fecha de obtención.

De los profesores se tiene su tipo, mientras que del personal en general se debe guardar su puesto. En el caso de los ayudantes, pueden ser de **investigador** o de **profesor**, almacenando el proyecto o materia en que laboran, respectivamente, así como sus horas asignadas. De los estudiantes debe tenerse almacenado el depto. al que pertenecen, así como distinguir si son de **posgrado** (almacenar el programa en que están) o **licenciatura**, donde debe guardarse su avance de créditos.



Atributo del supertipo que indica a que subtipos(s) pertenece

El mapeo de MERE a MR podemos hacerlo de cuatro maneras generales



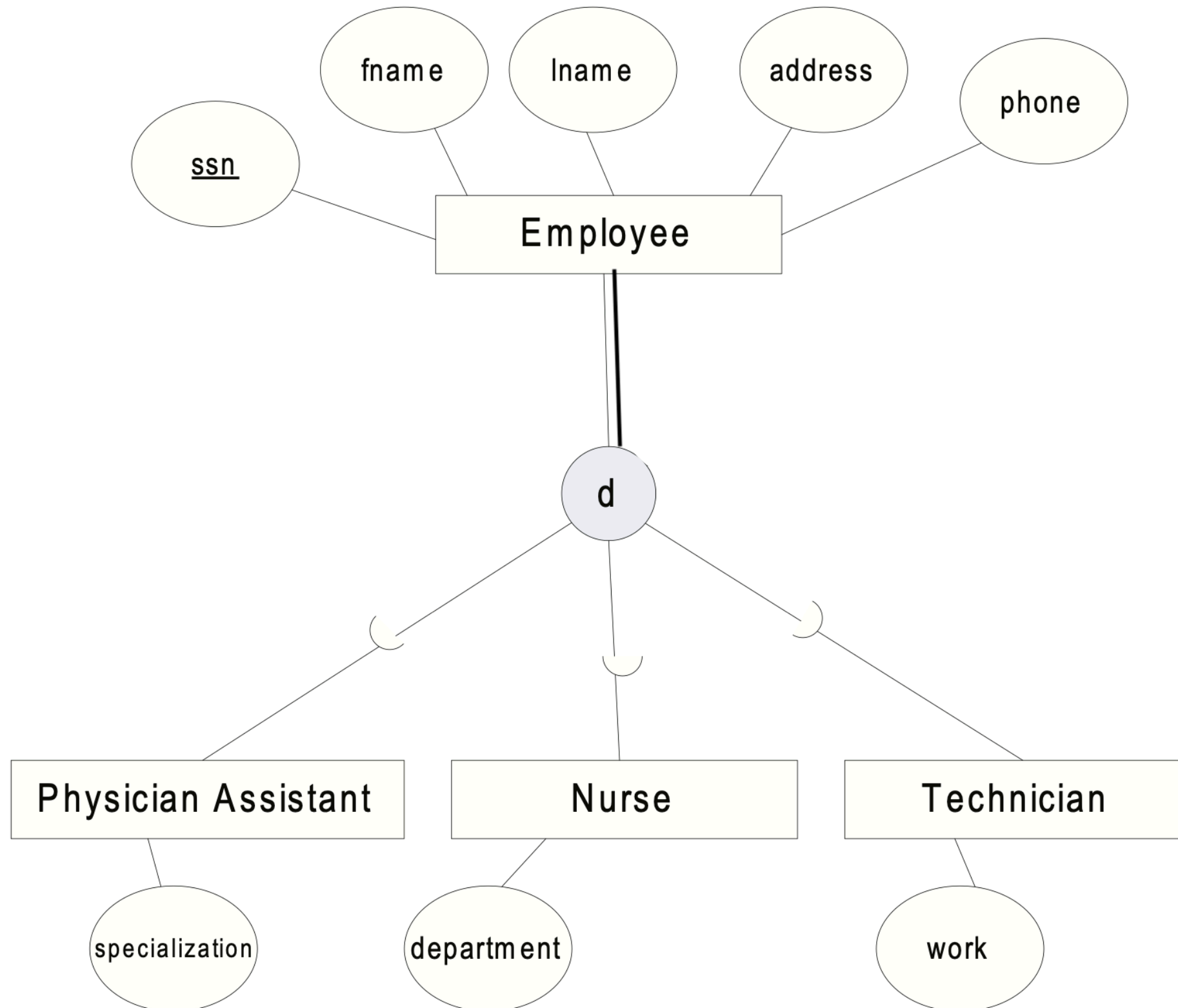
- **Crear una relación para el supertipo, incluyendo todos sus atributos**
- **Crear una relación independiente para cada subtipo, incluyendo la llave primaria del supertipo y los atributos del subtipo (si lo hay)**
- **Incluir atributo discriminante**

- **EMPLEADO: {ssn int (PK), nombre varchar(60), apellido varchar(120), direccion varchar(200), telefono int, tipo_trabajo char(1) ?? }**
- **ASISTENTE {ssn int (FK)(PK), especialidad varchar(100)}**
- **ENFERMERA {ssn int(FK)(PK), departamento varchar(60)}**
- **TECNICO: {ssn int(FK)(PK), oficio varchar(60)}**

¿Para qué restricciones trabaja bien?

- Crear una relación para cada uno de los subtipos. Incluir sus atributos propios (si los hay)**
- En cada relación de los subtipos, incluir todos los atributos del supertipo e indicar la llave primaria**

Transformación a MR

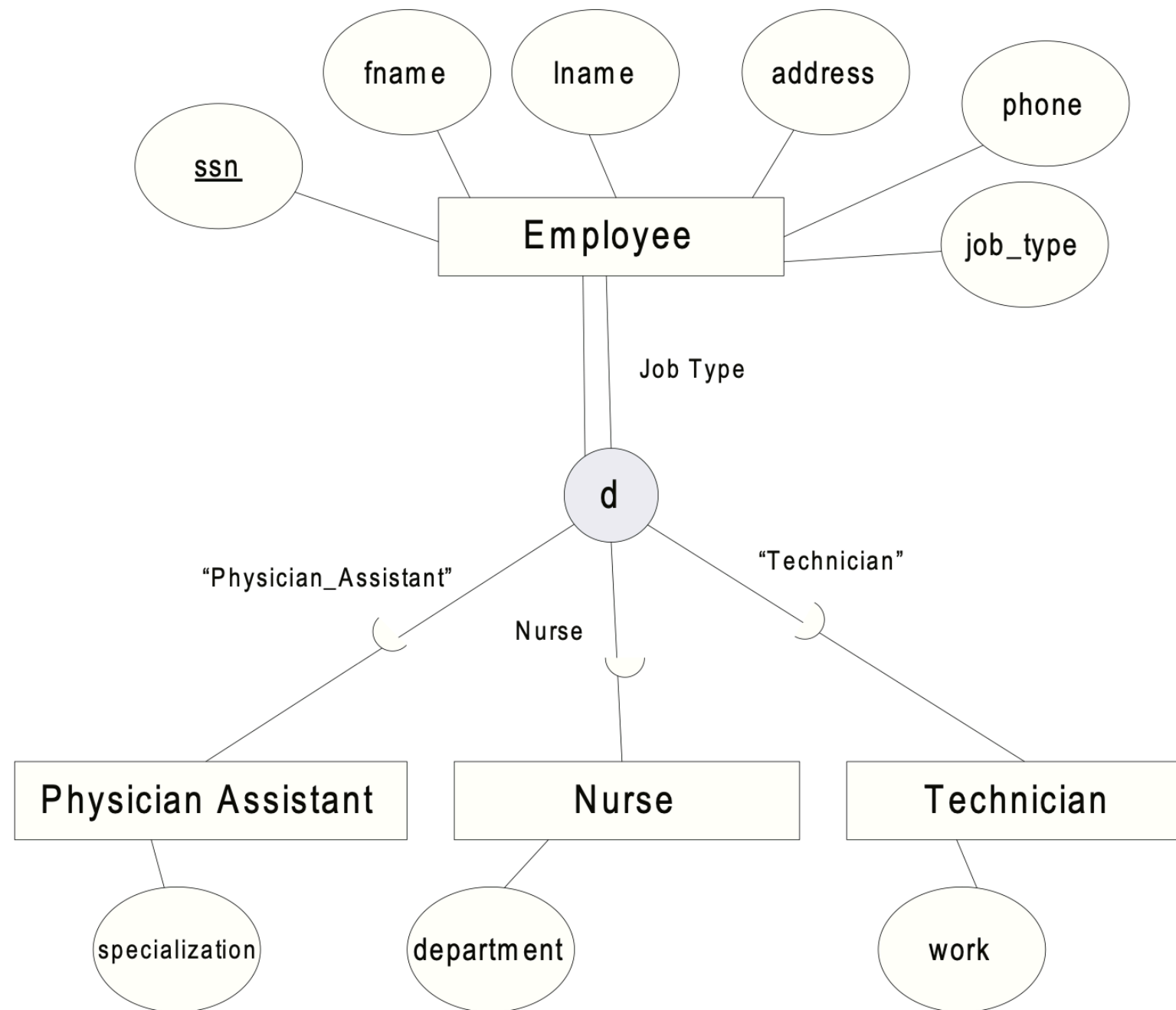


- **ASISTENTE:** {especialidad varchar(100), ssn int (PK), nombre varchar(60), apellido varchar(120), direccion varchar(200), telefono int}
- **ENFERMERA:** {departamento varchar(60), ssn int (PK), nombre varchar(60), apellido varchar(120), direccion varchar(200), telefono int}
- **TECNICO:** {oficio varchar(60), ssn int (PK), nombre varchar(60), apellido varchar(120), direccion varchar(200), telefono int}

**¿Para qué restricciones
trabaja bien?**

- Crear una relación para el supertipo, incluyendo todos sus atributos**
- En la nueva relación, agregar los atributos de los subtipos**
- Incluir atributo discriminante**

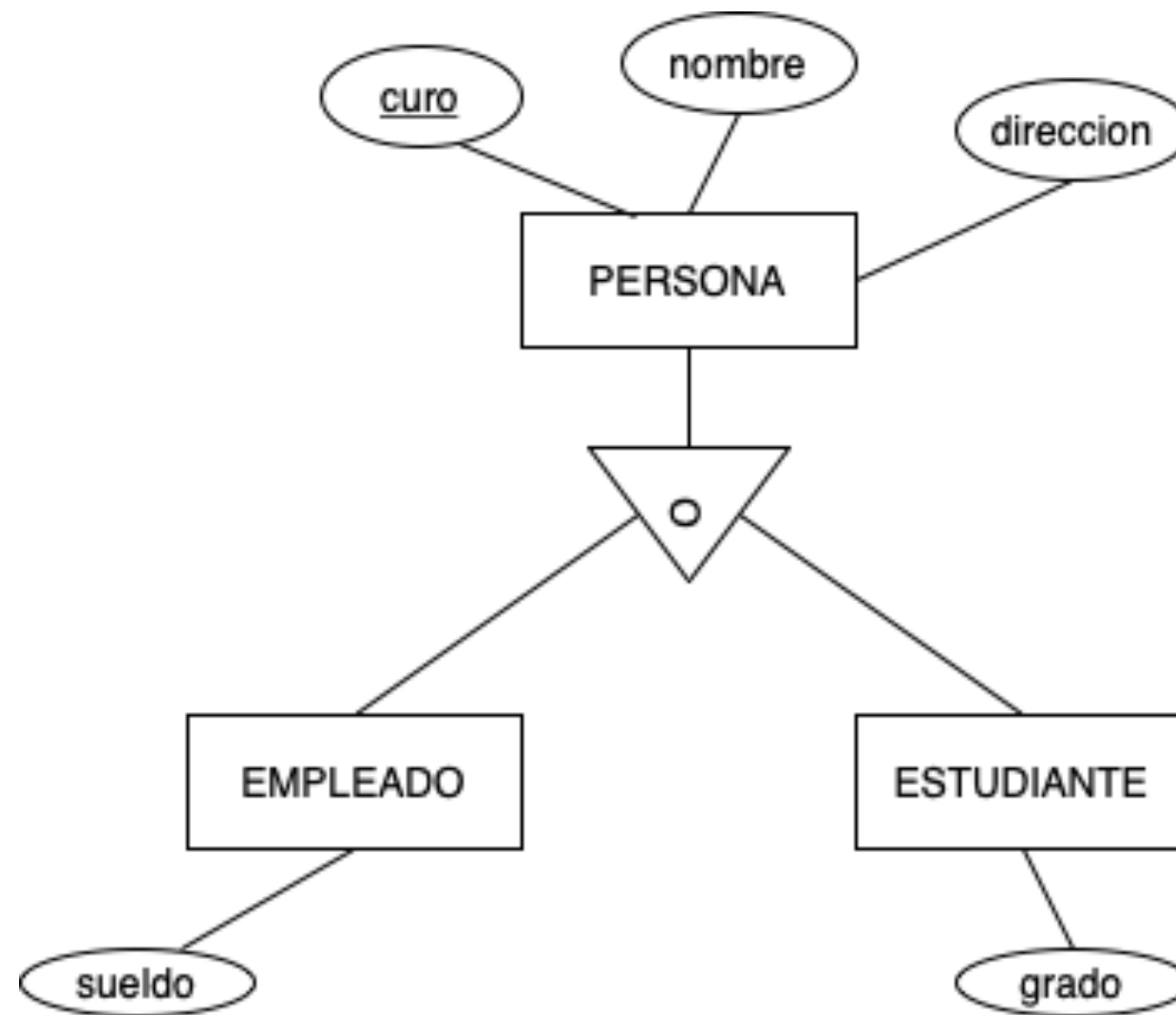
Transformación a MR



- EMPLEADO: {ssn int (PK), nombre varchar(60), apellido varchar(120), direccion varchar(200), telefono int, tipo_trabajo varchar(20) null, especialidad varchar(100) null, departamento varchar(60) null, oficio varchar(60) null}**

**¿Para qué restricciones
trabaja bien?**

- Crear una relación para el supertipo, incluyendo todos sus atributos**
- En la nueva relación, agregar los atributos de los subtipos, indicando para cada uno una bandera para hacer distinción**



- PERSONA: {curp varchar(18) (PK), nombre varchar(100), direccion varchar (200) es_estudiante char(1), grado varchar(30), es_trabajador char(1), sueldo int }**

**¿Para qué restricciones
trabaja bien?**

Proponer un requerimiento que:

- use los 3 tipos de cardinalidad
- al menos una generalización/especialización
- una agregación

**** entidad débil**

**** relación recursiva**