

### 3.3. Transformación de las relaciones

La transformación se realiza empleando las siguientes reglas:

- ✓ Toda **entidad** se transforma en una tabla.
- ✓ Todo **atributo** se transforma en columna dentro de la tabla.
- ✓ El identificador único de la entidad se convierte en **clave primaria**.
- ✓ Como las **relaciones** del modelo E/R no tienen equivalente en el modelo relacional, ya que sólo existen tablas y operaciones entre ellas, es necesario aplicar lo siguiente:
  - ✓ En las **relaciones M:N** se crea una nueva tabla que tendrá como clave primaria la concatenación de los atributos clave de las entidades que asocia y con los atributos propios de la relación si los hay. Esta tabla posee dos claves ajenas, una por cada entidad con la que está relacionada.
  - ✓ En las **relaciones 1:N** la entidad del lado N de la relación añade el conjunto de campos necesarios para incorporar a sus atributos la totalidad de la clave primaria de la entidad del lado 1, creando una clave ajena, de modo que se puedan relacionar ambas tablas mediante operadores relacionales. El nombre de la relación desaparece.
  - ✓ Las **relaciones 1:1** se transforman en función de las cardinalidades:
    - ✓ Cuando ambas entidades participan con cardinalidades (1,1) propagando cualquiera de los atributos identificadores y sus atributos asociados creando una única tabla con el conjunto de los atributos de ambas entidades. La clave primaria sería cualquiera de las dos.
    - ✓ Cuando ambas tablas tienen cardinalidades (0,1) crear una nueva tabla a partir de la relación con las dos claves de ambas.
    - ✓ Propagar la clave de la entidad con cardinalidad (1,1) a la entidad que tenga (0,1).

Veremos un ejemplo de aplicación de esas reglas:

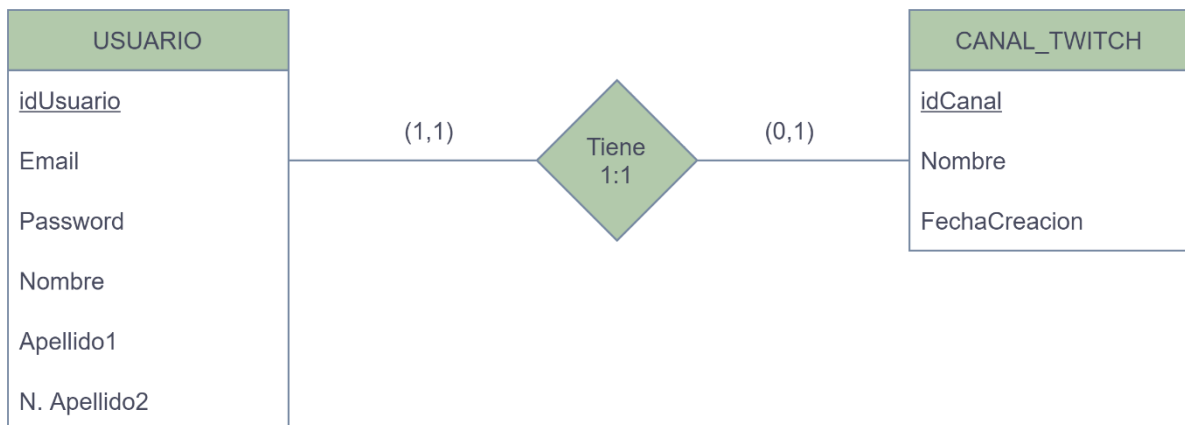


Además de las reglas de transformación que acabamos de ver y que se aplican con carácter general, existen otros aspectos a tener en cuenta a la hora de obtener un esquema relacional a partir de un modelo entidad-relación. Los veremos en los siguientes apartados.

#### 3.3.1. Transformación de relaciones 1:1

Para transformar las relaciones 1:1 lo que haremos será transformar la clave primaria de una de las entidades en clave foránea de la otra entidad. La participación de cada una de ellas nos ayudará a decidir cuál será la entidad que pasará su clave primaria a la otra entidad para evitar valores nulos.

- Participación (1,1) (0,1)



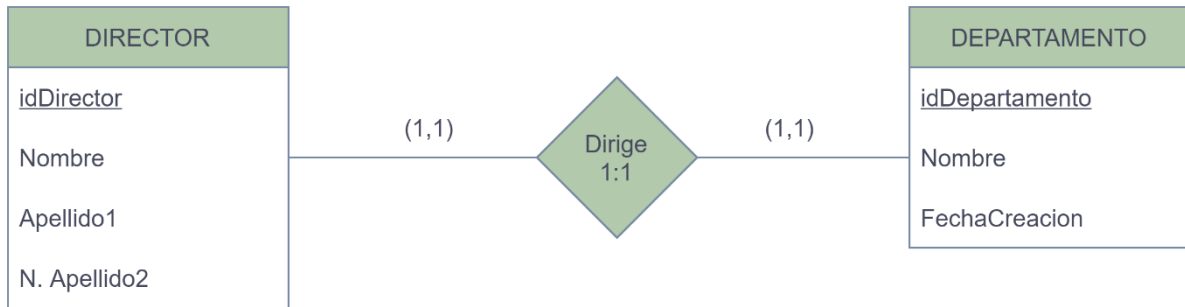
Como la participación de **USUARIO** es (1,1) y la de **CANAL\_TWITCH** es (0,1), la clave primaria de **USUARIO** se almacena en la relación **CANAL\_TWITCH** como clave foránea, quedando el esquema de la relación de la siguiente manera:

**USUARIO**(idUsuario, Email, Password, Nombre, Apellido1, N.Apellido2)

**CANAL\_TWITCH**(idCanal, Nombre, FechaCreacion, F. idUsuario  $\square$  **USUARIO**)

Aquí muestra el mejor caso cuando hay un (0x1/N), para evitar los nulos y se mas cómodo y visible.

- Participación (1,1) (1,1)

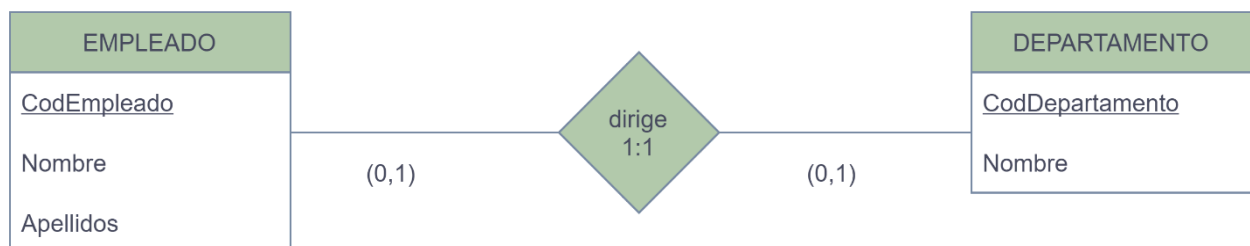


En este caso la participación de ambas entidades es (1,1) por tanto podemos:

- Almacenar la clave primaria de **DIRECTOR** en **DEPARTAMENTO** como clave foránea:  
**DIRECTOR**(idDirector, Nombre, Apellido1, N. Apellido2)  
**DEPARTAMENTO**(idDepartamento, Nombre, FechaCreacion, F. idDirector  $\square$  **DIRECTOR**)
- Almacenar la clave primaria de **DEPARTAMENTO** en **DIRECTOR** como clave foránea:  
**DEPARTAMENTO**(idDepartamento, Nombre, FechaCreacion)  
**DIRECTOR**(idDirector, Nombre, Apellido1, N. Apellido2, F. idDepartamento  $\square$  **DEPARTAMENTO**)
- Crear una sola relación, donde la clave primaria puede ser cualquiera de las claves primarias de las dos entidades:  
**DIRECTOR\_DEPARTAMENTO**(idDirector, idDepartamento, NombreDirector, Apellido1, N. Apellido2, NombreDepartamento, FechaCreacion)  
 O bien:  
**DIRECTOR\_DEPARTAMENTO**(idDirector, idDepartamento, NombreDirector, Apellido1, N. Apellido2, NombreDepartamento, FechaCreacion)

- Participación (0,1)(0,1)

La teoría dice que generaremos una nueva relación para evitar valores nulos en las claves ajenas o foráneas, teniendo como clave primaria la concatenación de las claves primarias de las entidades asociadas



**EMPLEADO**(CodEmpleado, Nombre, Apellidos)

**DEPARTAMENTO**(CodDepartamento, Nombre)

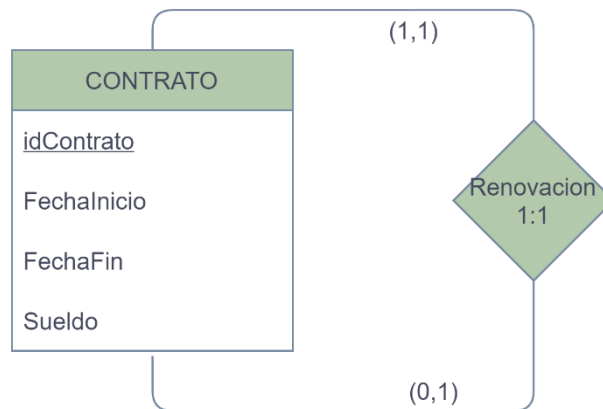
**EMPLEADO\_DIRIGE\_DEPARTAMENTO**(F.CodEmpleado  $\square$  **EMPLEADO**, F.CodDepartamento  $\square$  **DEPARTAMENTO**)

En la práctica, escoges la tabla que vaya a tener el menor número de filas (en el ejemplo parece claro que será departamento) y la relaciones con el empleado admitiendo valores nulos.

**EMPLEADO**(CodEmpleado, Nombre, Apellidos)

**DEPARTAMENTO**(CodDepartamento, Nombre, N.Director  $\square$  EMPLEADO )

- Relaciones reflexivas con cardinalidad 1:1

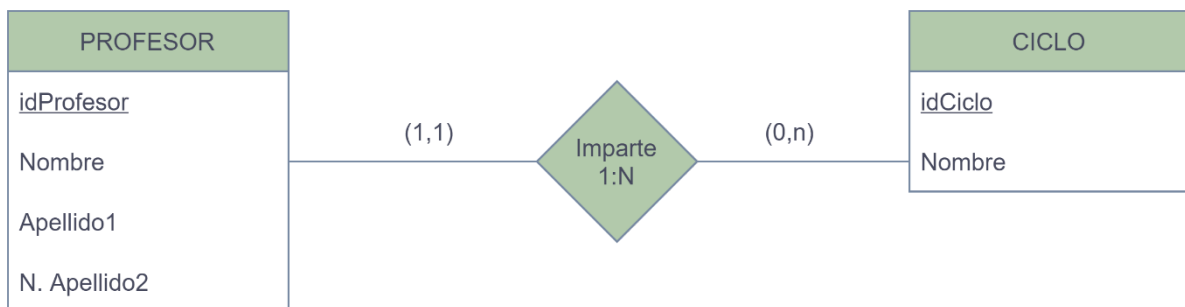


En este caso la clave primaria se vuelve a guardar en la misma relación como clave foránea

**CONTRATO**( idContrato, FechaInicio, FechaFin, Sueldo, F. idContratoAnterior  $\square$  **CONTRATO**)

### 3.3.2. Transformación de relaciones 1:N

Cuando nos encontramos con un caso de este tipo, haremos que la clave primaria de la entidad que participa con cardinalidad 1 pase a formar parte de la entidad que participa con cardinalidad N como clave foránea

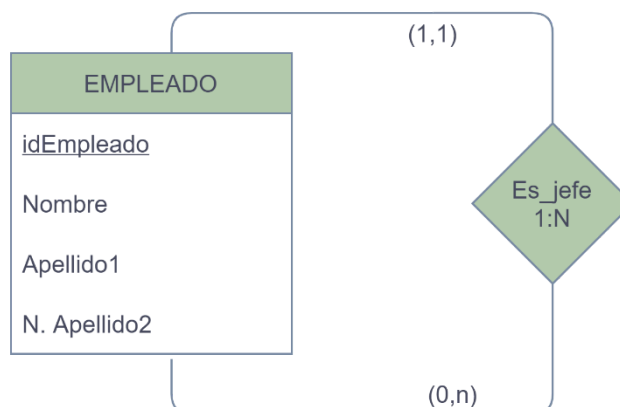


En este caso la entidad **PROFESOR** es la que participa con cardinalidad 1, por tanto pasará a **CICLO** como clave ajena o foránea:

**PROFESOR**(idProfesor, Nombre, Apellido1, Apellido2)

**CICLO**(idCiclo, Nombre, F. idProfesor  $\square$  **PROFESOR**)

- Relaciones reflexivas con 1:N



En este caso la clave primaria se almacena en la misma tabla como clave foránea:

**EMPLEADO**( idEmpleado,Nombre,Apellido1,N. Apellido2,F. idJefe  $\square$  **EMPLEADO**)

### Excepciones en relaciones 1-N

En los siguientes casos interesa más crear una nueva tabla a partir de la relación como en el caso de correspondencias M:N:

1. Cuando el número de ocurrencias de la entidad que propaga la clave es muy pequeño y cabe la posibilidad de que al propagar la clave quedan muchos valores repetidos o nulos.
2. Cuando se prevea que en el futuro se puede convertir en una relación M:N
3. Cuando la relación tenga atributos propios. En algunos casos se pueden migrar estos atributos junto con la clave pero, en general, se crea una nueva tabla.

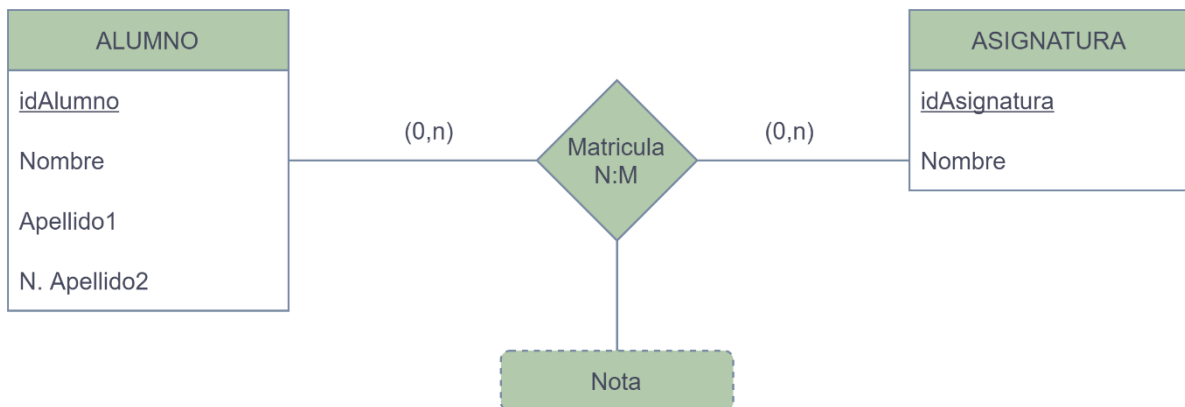
En este caso mejor FK con nula ->

1) Cuando el número de empleados que están afiliados es muy pequeño.		<b>SINDICATO</b> :( <u>IDSindicato</u> , Nombre, ..... ) Afiliar: ( <u>IDSindicato + IDEmpleado</u> ) <b>EMPLEADO</b> : ( <u>IDEmpleado</u> , Nombre,.....)
2) Si en el futuro una misma pieza puede ser suministrada por distintos proveedores.		<b>PROVEEDOR</b> :( <u>IDProveedor</u> , Nombre,..IDPieza) Suministra: ( <u>IDProveedor + IDPieza</u> ) <b>PIEZA</b> : ( <u>IDPieza</u> , Descripción,.....)
3) Un alumno puede retirar varios ejemplares. No se recoge la dimensión temporal		<b>ALUMNO</b> : ( <u>IDAlumno</u> , Nombre.....) Prestar: ( <u>IDAlumno + IDEjemplar</u> , F_Ini, F_Fin) <b>EJEMPLAR</b> : ( <u>IDEjemplar</u> , Título,.....)

3.3.3.

### Relaciones con cardinalidad N:M

Las relaciones con cardinalidad N:M generarán una nueva tabla donde se almacenarán las claves primarias de las dos entidades que participan en la relación, siendo las claves primarias de cada una de ellas también claves primarias de la nueva tabla. Si la relación contiene algún atributo, éste también se añade a la nueva tabla.



**ALUMNO**(idAlumno, Nombre, Apellido1,N. Apellido2)

**ASIGNATURA**(idAsignatura,Nombre)

**ALUMNO\_MATRICULA\_ASIGNATURA**(F.idAlumno  $\square$  **ALUMNO**,F.idAsignatura  $\square$  **ASIGNATURA**, Nota)

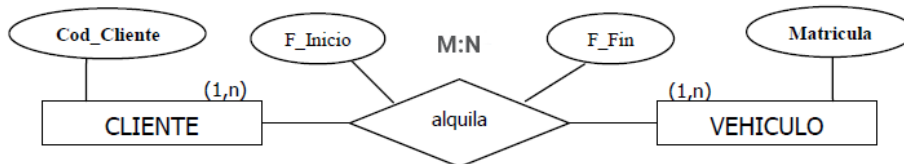
Si en este ejemplo, nos dijeran que el alumno se puede matricular más de una vez o que se quiere guardar un histórico de las matrículas de un alumno, deberíamos también tener un atributo **Fecha** de la relación. Este atributo pasará a formar parte de la clave primaria de la relación **ALUMNO\_MATRICULA\_ASIGNATURA**:

**ALUMNO\_MATRICULA\_ASIGNATURA**(F.idAlumno  $\square$  **Alumno**,F.idAsignatura  $\square$  **Asignatura**, Fecha, Nota)

## Relaciones N-M con dimensión temporal

3.3.4.

Ejemplo: Relación entre los clientes que alquilan los vehículos de una empresa de alquileres. Recogemos los alquileres realizados a nuestros clientes a lo largo del tiempo.

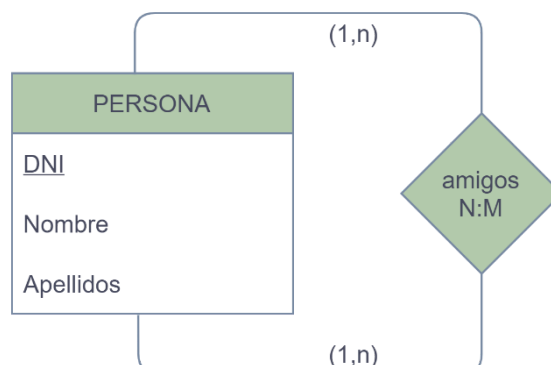


alquila ( Cod\_Cliente, Matricula, F\_Inicio, F\_Fin)

CLIENTE ( Cod\_Cliente, .....)

VEHICULO ( Matricula, .....)

## Relaciones reflexivas con cardinalidad N:M



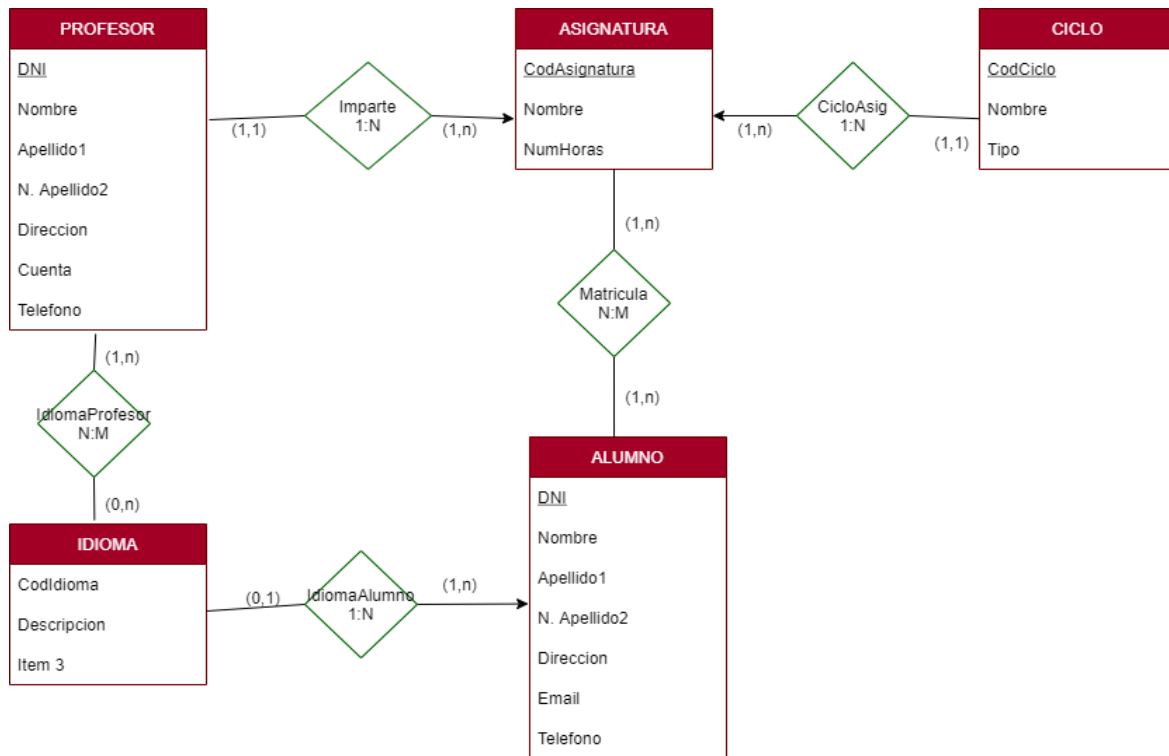
Necesita crear una tabla de la relación  
Por ser una NxM

**PERSONA**(DNI,Nombre,Apellidos)

**PERSONA\_AMIGO**(F.DNI\_Amigo1 **AMIGO**, F.DNI\_Amigo2 **AMIGO**)

## Ejercicio resuelto 1

Partiendo del siguiente modelo conceptual, vamos a obtener su esquema relacional



### Solución

1º Transformamos directamente las entidades con sus claves primarias y atributos:

PROFESOR(DNI, Nombre, Apellido1, N.Apellido2, Direccion, Cuenta, Telefono)

ASIGNATURA(CodAsignatura, Nombre, NumHoras)

CICLO(CodCiclo, Nombre, Tipo)

IDIOMA(CodIdioma, Descripcion)

ALUMNO(DNI, Nombre, Apellido1, N.Apellido2, Direccion, Email, Telefono)

2º Nos fijamos en las multiplicidades de sus relaciones:

Entre **PROFESOR** y **ASIGNATURA** tenemos una relación 1:N, **PROFESOR** es la que participa con 1, por tanto añadiremos una clave ajena en **ASIGNATURA** que sea la clave primaria de profesor

PROFESOR(DNI, Nombre, Apellido1, N.Apellido2, Direccion, Cuenta, Telefono)

ASIGNATURA(CodAsignatura, Nombre, NumHoras, F.DNI\_Profesor  $\square$  PROFESOR)

CICLO(CodCiclo, Nombre, Tipo)

IDIOMA(CodIdioma, Descripcion)

ALUMNO(DNI, Nombre, Apellido1, N.Apellido2, Direccion, Email, Telefono)

La relación entre **ASIGNATURA** y **CICLO** es de multiplicidad 1:N y es **CICLO** quien participa con 1, por eso pasaremos su clave principal a **ASIGNATURA** como clave ajena. Como la relación a la que apunta la clave ajena debe estar declarada con anterioridad, debemos declarar **CICLO** antes que **ASIGNATURA**

PROFESOR(DNI, Nombre, Apellido1, N.Apellido2, Direccion, Cuenta, Telefono)  
 CICLO(CodCiclo, Nombre, Tipo)  
 ASIGNATURA(CodAsignatura, Nombre, NumHoras, F.DNI\_Profesor  $\square$  PROFESOR, F.CodCiclo  $\square$  CICLO)  
 IDIOMA(CodIdioma, Descripcion)  
 ALUMNO(DNI, Nombre, Apellido1, N.Apellido2, Direccion, Email, Telefono)

La relación entre **ALUMNO** e **IDIOMA** es 1:N y es **IDIOMA** la relación que tiene el 1, por eso **IDIOMA** genera en **ALUMNO** una clave ajena. Además la cardinalidad mínima es 0, esto significa que puede haber alumnos que no hablen ningún idioma, por lo que la clave ajena puede aceptar nulos, por tanto a la clave ajena le añadiremos la letra N

PROFESOR(DNI, Nombre, Apellido1, N.Apellido2, Direccion, Cuenta, Telefono)  
 CICLO(CodCiclo, Nombre, Tipo)  
 ASIGNATURA(CodAsignatura, Nombre, NumHoras, F.DNI\_Profesor  $\square$  PROFESOR, F.CodCiclo  $\square$  CICLO)  
 IDIOMA(CodIdioma, Descripcion)  
 ALUMNO(DNI, Nombre, Apellido1, N.Apellido2, Direccion, Email, Telefono, FN.CodIdioma  $\square$  IDIOMA)

La relación **Matrícula** entre **ALUMNO** y **ASIGNATURA** es de multiplicidad N:M, por lo que generará una nueva relación llamada **MATRICULA**, cuya clave primaria estará formada por las claves primarias de **ALUMNO** y **ASIGNATURA** que también actuarán como claves foráneas

PROFESOR(DNI, Nombre, Apellido1, N.Apellido2, Direccion, Cuenta, Telefono)  
 CICLO(CodCiclo, Nombre, Tipo)  
 ASIGNATURA(CodAsignatura, Nombre, NumHoras, F.DNI\_Profesor  $\square$  PROFESOR, F.CodCiclo  $\square$  CICLO)  
 IDIOMA(CodIdioma, Descripcion)  
 ALUMNO(DNI, Nombre, Apellido1, N.Apellido2, Direccion, Email, Telefono)  
 MATRICULA(F.DNI\_Alumno  $\square$  ALUMNO, F.CodASignatura  $\square$  ASIGNATURA)

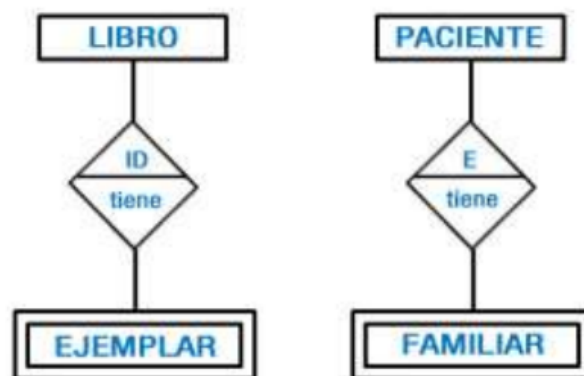
Ahora vamos a la relación **IdiomaProfesor** entre **PROFESOR** e **IDIOMA**, también tiene multiplicidad N:M, por tanto tendremos que crear una nueva relación **IDIOMAPROFESOR** que tendrá como clave primaria las dos claves primarias de **IDIOMA** y **PROFESOR** que también serán claves ajenas. También vemos que puede que el **PROFESOR** no hable ningún **IDIOMA**, pero no es necesario indicarlo con la restricción NULL porque en el caso de relaciones N:M, si un **PROFESOR** no habla ningún **IDIOMA** no aparecerá en esta relación.

PROFESOR(DNI, Nombre, Apellido1, N.Apellido2, Direccion, Cuenta, Telefono)  
 CICLO(CodCiclo, Nombre, Tipo)  
 ASIGNATURA(CodAsignatura, Nombre, NumHoras, F.DNI\_Profesor  $\square$  PROFESOR, F.CodCiclo  $\square$  CICLO)  
 IDIOMA(CodIdioma, Descripcion)  
 ALUMNO(DNI, Nombre, Apellido1, N.Apellido2, Direccion, Email, Telefono)  
 IDIOMAPROFESOR(F.CodIdioma  $\square$  IDIOMA, F.DNI  $\square$  PROFESOR)

### 3.3.4. Entidades fuertes y entidades débiles

Como hemos dicho por cada entidad fuerte de un diagrama E-R crearemos una nueva tabla en el esquema relacional con tantas columnas como atributos posea la entidad. Las entidades débiles las transformaremos de manera diferente dependiendo de su tipo de dependencia:

- Dependencia de existencia: las tratamos como entidades fuertes
- Dependencia de identificación: crearemos una nueva tabla que tendrá como clave primaria la concatenación de la clave primaria de la entidad padre y la clave primaria de la entidad débil. Además tendremos la clave primaria de la entidad padre como clave foránea de la entidad débil



#### Dependencia de identificación:

**LIBRO**(ISBN, Titulo, NumPaginas)

**EJEMPLAR**(NumPublicacion, F.ISBN LIBRO, FechaPublicacion)

#### Dependencia de existencia (se transforma como una relación 1 a n “normal”):

**PACIENTE**(DOCUMENTO, Nombre, Apellido)

**FAMILIAR**(DOCUMENTO, Nombre, Apellido, F.DOCUMENTO PACIENTE)