


BASES DE DATOS

PASO DE ER A TABLAS



ESCUELA DE INGENIERÍA INFORMÁTICA

Contenidos

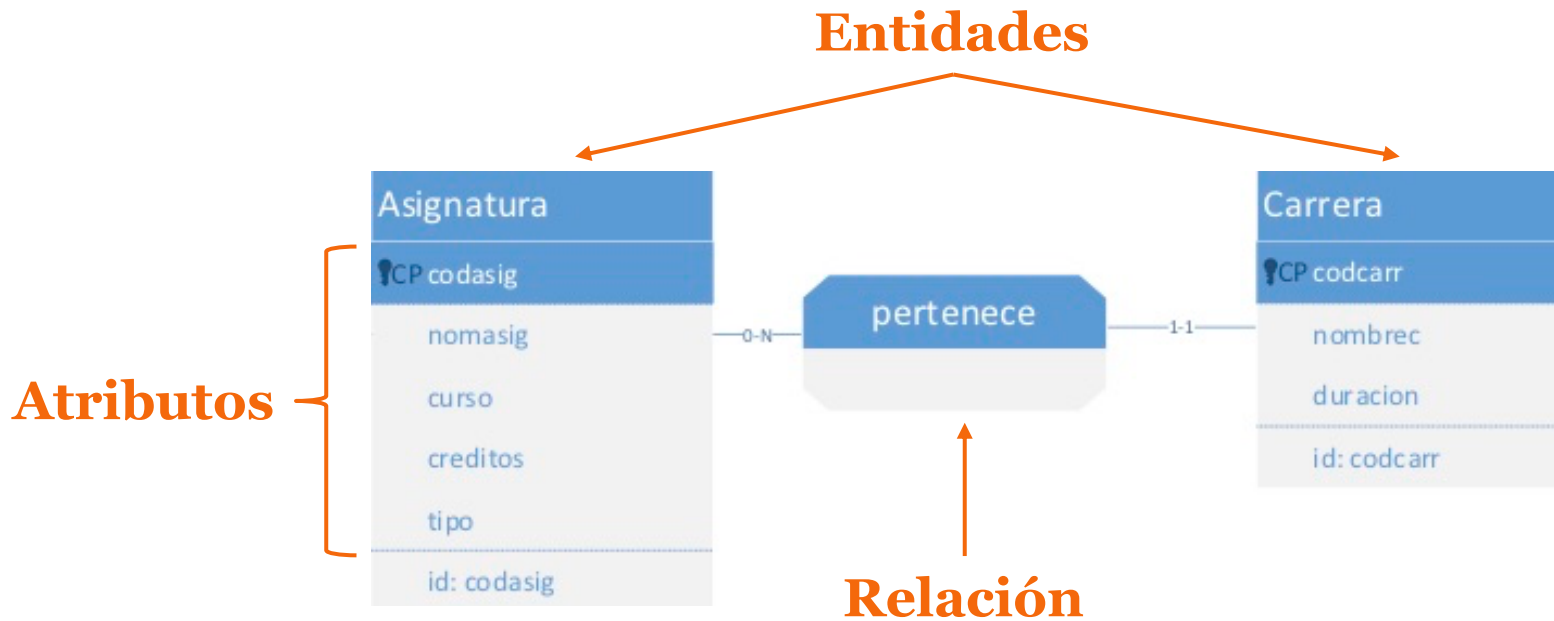
-  1. Introducción al modelo ER
- 2. Conversión de elementos básicos
- 3. Restricciones
- 4. Otros elementos

Modelo ER

- ∞ El modelo entidad-relación (ER) permite representar la estructura de una base de datos relacional mediante un diagrama compuesto de entidades y relaciones, que a su vez pueden contener atributos.
 - **Entidad:** objeto de la base de datos que se representa con un rectángulo.
 - **Relación:** dependencia entre entidades que se representa con un octógono (rectángulo con las esquinas cortadas), o un rombo, que se une a las entidades mediante líneas.
 - **Atributo:** característica de una entidad o relación.

Modelo ER

∞ Ejemplo:



Modelo ER

∞ Los diferentes elementos que componen el modelo ER se convierten en un **conjunto de tablas** que se implementan en el gestor utilizando la sentencia **CREATE TABLE**.

- Tabla: entidad o relación del modelo ER.
- Campo: atributo del modelo ER.
- Restricciones:
 - Clave primaria: PRIMARY KEY
 - Clave externa: FOREIGN KEY
 - Obligatoriedad (atributos no vacíos): NOT NULL
 - Unicidad (clave candidata): UNIQUE
 - Verificación (dominio de un atributo): CHECK

CREATE TABLE

∞ Sintaxis de la sentencia para crear una tabla.

```
CREATE TABLE nombre_tabla (  
    campo tipo_campo [NOT NULL]  
    [,campo tipo_campo [NOT NULL]]  
    [,CONSTRAINT nombre_clave_primaria PRIMARY KEY (campo [,campo])]  
    [,CONSTRAINT nombre_clave_externa FOREIGN KEY (campo [,campo])  
        REFERENCES tabla_referencia (campo [,campo])]  
    [,CONSTRAINT nombre_clave_candidata UNIQUE (campo [,campo])]  
    [,CONSTRAINT nombre_check CHECK (condición)]  
);
```

Contenidos

∞ 1. Introducción al modelo ER

 ∞ 2. Conversión de elementos básicos

∞ 3. Restricciones

∞ 4. Otros elementos

Entidad

∞ Crear una tabla para cada entidad del modelo ER.

- Nombre de la tabla: nombre de la entidad.
- Campos: atributos de la entidad.
- Clave primaria: identificada como CP.

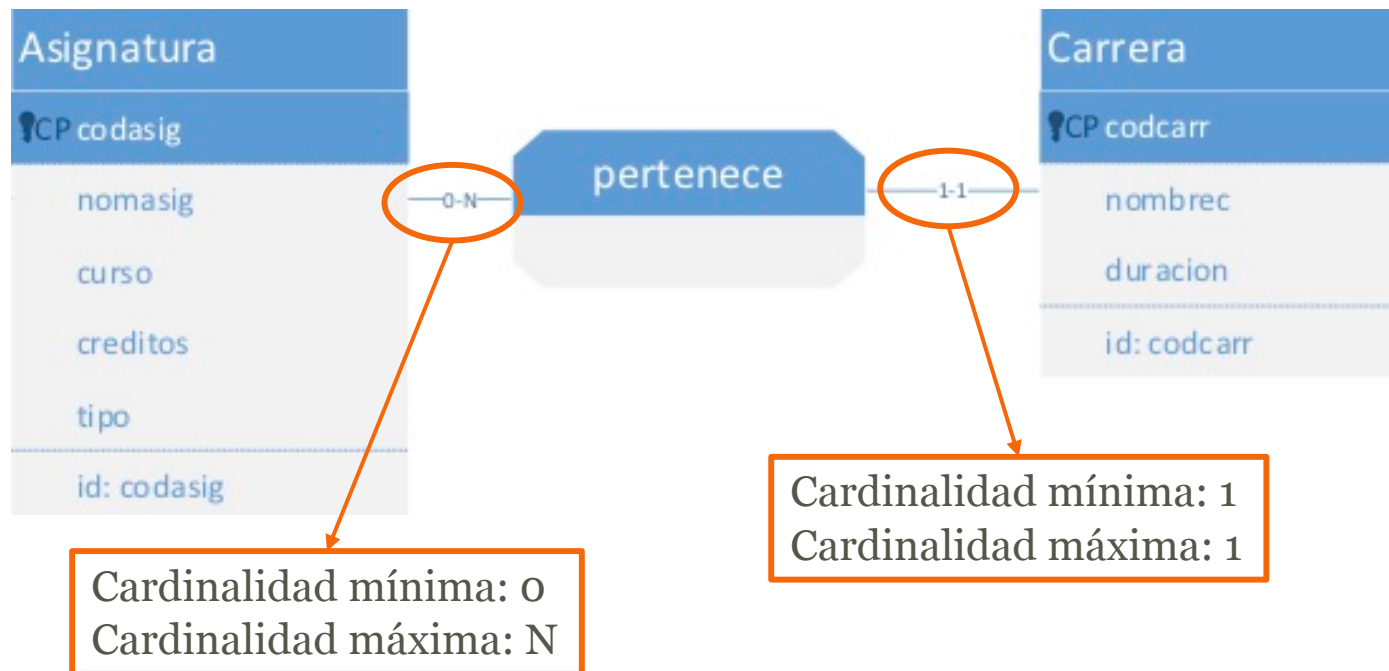
∞ Ejemplo:

Carrera	
CP	codcarr
	nombrec
	duracion
	id: codcarr

CREATE TABLE CARRERA
(
 CODCARR varchar(8),
 NOMBREC varchar(20),
 DURACION decimal(1,0),
 CONSTRAINT PK_CARRERA PRIMARY KEY (CODCARR)
);

Relaciones

- Cardinalidad: número de veces que una instancia (tupla, fila) de una tabla puede asociarse a una instancia de otra tabla con la que está relacionada.
- Ejemplo:

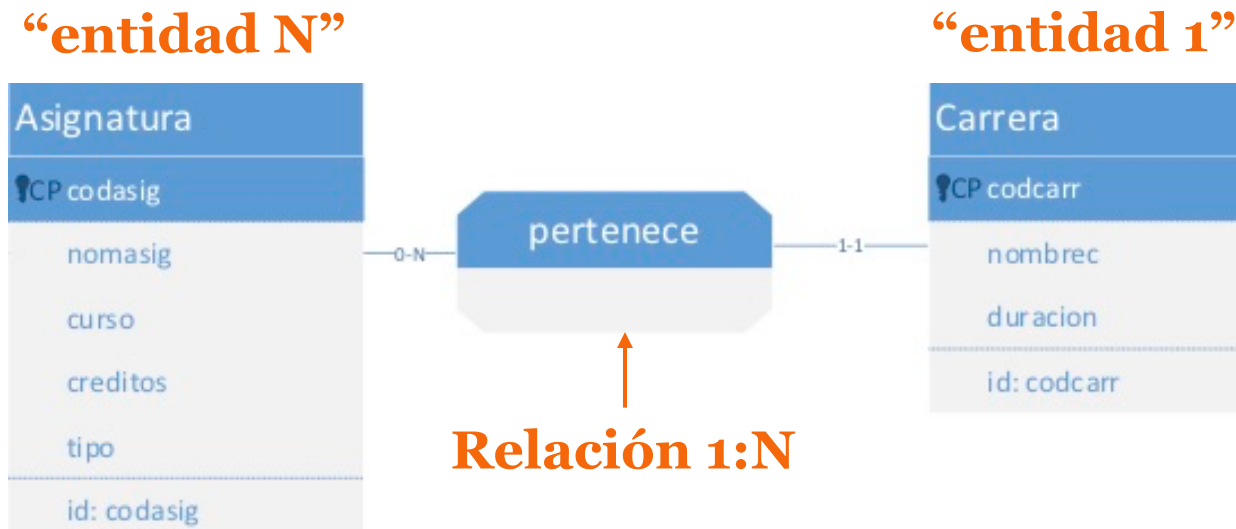


Relaciones

- ∞ El tipo de una relación viene determinado por la **cardinalidad máxima** de las entidades participantes:
 - Uno a muchos (1:N)
 - Muchos a muchos (N:N)
- ∞ El proceso de conversión de una relación depende de su tipo.

Relación 1:N

- ∞ Añadir el atributo(s) de la clave primaria de la “entidad 1” a la tabla de la “entidad N”.
 - Clave externa: el atributo(s) añadido referencia a la tabla de la “entidad 1”.
- ∞ Ejemplo:



Relación 1:N

∞ Ejemplo (continuación):

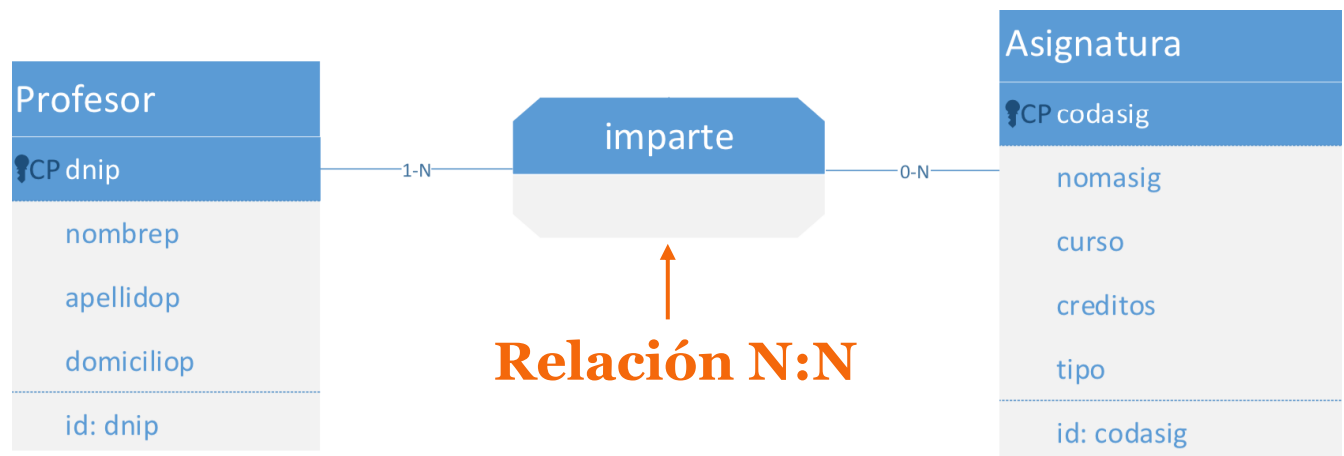
```
CREATE TABLE ASIGNATURA
(
    CODASIG varchar(8),
    NOMASIG varchar(20),
    CURSO decimal(1,0),
    CREDITOS decimal(2,0),
    TIPO varchar(20),
    CODCARR varchar(8) NOT NULL,
    CONSTRAINT PK_ASIGNATURA PRIMARY KEY (CODASIG),
    CONSTRAINT FK_ASIGNATURA_CARRERA FOREIGN KEY (CODCARR)
        REFERENCES CARRERA (CODCARR)
);
```

Relación N:N

∞ Crear una nueva tabla.

- Nombre de la tabla: nombre de la relación en el modelo ER.
- Campos: atributo(s) de las claves primarias de las entidades participantes + atributos propios de la relación (si los hay).
- Clave primaria: compuesta por los atributos de las claves primarias de las entidades participantes.
- Claves externas: los atributos añadidos referencian a las tablas de las entidades correspondientes.

∞ Ejemplo:



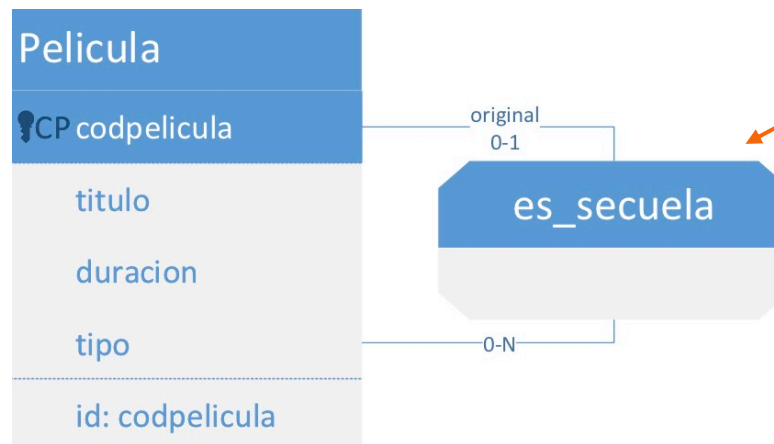
Relación N:N

∞ Ejemplo (continuación):

```
CREATE TABLE IMPARTE
(
    CODASIG varchar(8),
    DNIP varchar(8),
    CONSTRAINT PK_IMPARTE PRIMARY KEY (CODASIG, DNIP),
    CONSTRAINT FK_IMPARTE_ASIGNATURA FOREIGN KEY (CODASIG)
        REFERENCES ASIGNATURA (CODASIG),
    CONSTRAINT FK_IMPARTE_PROFESOR FOREIGN KEY (DNIP)
        REFERENCES PROFESOR (DNIP)
);
```

Relación reflexiva

- Relación en la que los dos extremos pertenecen a la misma entidad.
- Se resuelve en función de su tipo (1:N o N:N), siguiendo los pasos descritos anteriormente.
- Ejemplo:



Relación reflexiva 1:N

Relación reflexiva

∞ Ejemplo (continuación):

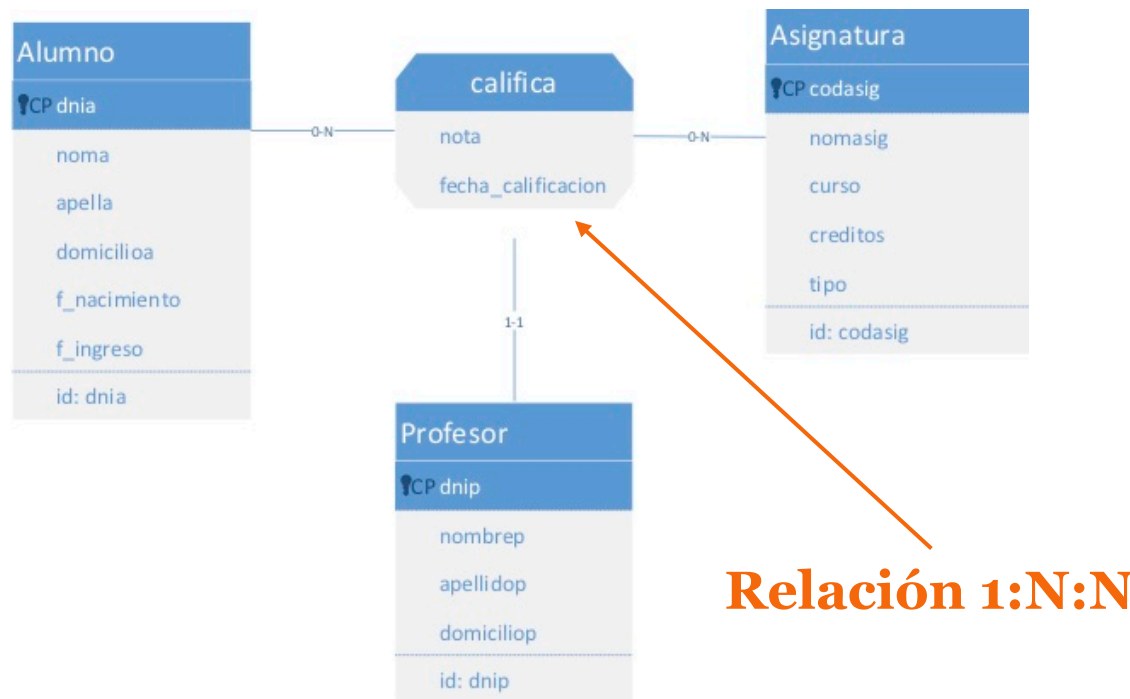
```
CREATE TABLE PELICULA
(
    CODPELICULA varchar(4),
    TITULO varchar(20),
    DURACION decimal(2,0),
    TIPO varchar(20),
    CODPELICULA_ORIGINAL varchar(4),
    CONSTRAINT PK_PELICULA PRIMARY KEY (CODPELICULA),
    CONSTRAINT FK_PELICULA_PELICULA_ORIGINAL FOREIGN KEY
        (CODPELICULA_ORIGINAL) REFERENCES PELICULA (CODPELICULA)
);
```


Relación ternaria

- ∞ Relación compuesta por tres entidades.
- ∞ Se resuelve en función de su tipo ($1:1:N$, $1:N:N$ o $N:N:N$), creando siempre una nueva tabla.
 - Nombre de la tabla: nombre de la relación en el modelo ER.
 - Campos: atributo(s) de las claves primarias de las entidades participantes + atributos propios de la relación (si los hay).
 - Clave primaria: hay una clave candidata por cada “entidad 1”, cuyos atributos se componen de la unión de las claves primarias de las otras dos entidades.
 - Claves externas: los atributos añadidos referencian a las tablas de las entidades correspondientes.

Relación ternaria

∞ Ejemplo:




Relación ternaria

☞ Ejemplo (continuación):

```
CREATE TABLE CALIFICA
(
    DNIA varchar(8),
    CODASIG varchar(8),
    DNIP varchar(8) NOT NULL,
    NOTA decimal(2,0),
    FECHA_CALIFICACION date,
    CONSTRAINT PK_CALIFICA PRIMARY KEY (DNIA, CODASIG),
    CONSTRAINT FK_CALIFICA_ALUMNO FOREIGN KEY (DNIA) REFERENCES ALUMNO
        (DNIA),
    CONSTRAINT FK_CALIFICA_ASIGNATURA FOREIGN KEY (CODASIG) REFERENCES
        ASIGNATURA (CODASIG),
    CONSTRAINT FK_CALIFICA_PROFESOR FOREIGN KEY (DNIP) REFERENCES PROFESOR
        (DNIP)
);
```

Contenidos

- 1. Introducción al modelo ER
- 2. Conversión de elementos básicos
-  3. Restricciones
- 4. Otros elementos

Restricción NOT NULL

⌘ Obligatoriedad (atributos no vacíos)

- Por defecto, un campo puede contener valores NULL.
- La restricción **NOT NULL** obliga a un campo a no permitir valores NULL.
 - No se puede insertar (o actualizar) una instancia sin especificar un valor para este campo.
- ¿Cuándo se utiliza?
 - En las relaciones 1:N (o equivalentes), cuando la cardinalidad mínima de la “entidad 1” es también 1 (excepto si el campo forma parte de la clave primaria). Ejemplo (ver transparencia 11):
CODCARR varchar(8) NOT NULL
 - En los casos en los que se indique explícitamente en la descripción del problema (p. ej., “es un campo obligatorio”, “no admite valores nulos”).

Restricción UNIQUE

∞ Unicidad (clave candidata)

- Una clave candidata es una clave única que puede ser utilizada como clave principal.
- La restricción **UNIQUE** garantiza que todos los valores de un campo(s) son diferentes; es decir, no se admiten repetidos.
- ¿Cuándo se utiliza?
 - En los casos en los que por la naturaleza del campo no existan valores repetidos, excepto si el campo forma parte de una clave primaria (p. ej., pasaporte, DNI).
 - En los casos en los que se indique explícitamente en la descripción del problema. Ejemplo (ver transparencia 11): “No existen asignaturas con el mismo nombre”

CONSTRAINT UQ_ASIGNATURA_NOMASIG UNIQUE (NOMASIG)

Restricción UNIQUE

- ¿Cuándo se utiliza? (continuación)
 - En los atributos de la clave externa definida para resolver una relación 1:1 (caso particular de las relaciones 1:N que se resuelve tratando, indistintamente, una de las dos entidades participantes como “entidad N”).


Restricción CHECK

∞ Verificación (dominio de un atributo)

- Es posible limitar los valores en ciertos campos de una tabla.
- La restricción **CHECK** indica el rango de valores admitidos en un campo.
- Es posible combinarla con la restricción NOT NULL.
- ¿Cuándo se utiliza?
 - En los casos en los que se indique explícitamente en la descripción del problema. Ejemplo (ver transparencia 11): “Las asignaturas pueden ser: obligatorias, optativas, o de libre configuración”

```
TIPO varchar(20) NOT NULL,  
CONSTRAINT CK_ASIGNATURA_TIPO CHECK (TIPO IN ('obligatoria',  
        'optativa', 'libre configuracion'))
```


Contenidos

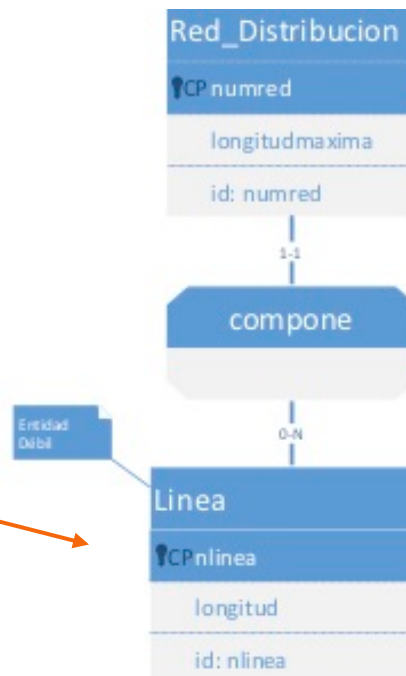
- 1. Introducción al modelo ER
- 2. Conversión de elementos básicos
- 3. Restricciones
-  4. Otros elementos

Entidad débil

- Una **entidad débil** es aquella que no puede existir sin participar en una relación; es decir, no puede ser identificada de forma única por sus atributos.
 - Debe utilizar una clave externa para definir la clave principal.

Ejemplo:

Entidad débil



El hecho de necesitar la clave primaria de “Red_Distribucion” en la tabla de la entidad “Linea” ya implementa la relación 1:N “compone”.

Entidad débil

∞ Ejemplo (continuación):

```
CREATE TABLE RED_DISTRIBUCION
(
    NUMRED decimal(3,0),
    ENOMBRE varchar(15) NOT NULL,
    LONGITUDMAXIMA decimal(10,2),
    CONSTRAINT PK_RED_DISTRIBUCION PRIMARY KEY (NUMRED),
    CONSTRAINT FK_RED_DISTRIBUCION_ESTACION FOREIGN KEY (ENOMBRE)
        REFERENCES ESTACION (ENOMBRE)
);
```

Entidad débil

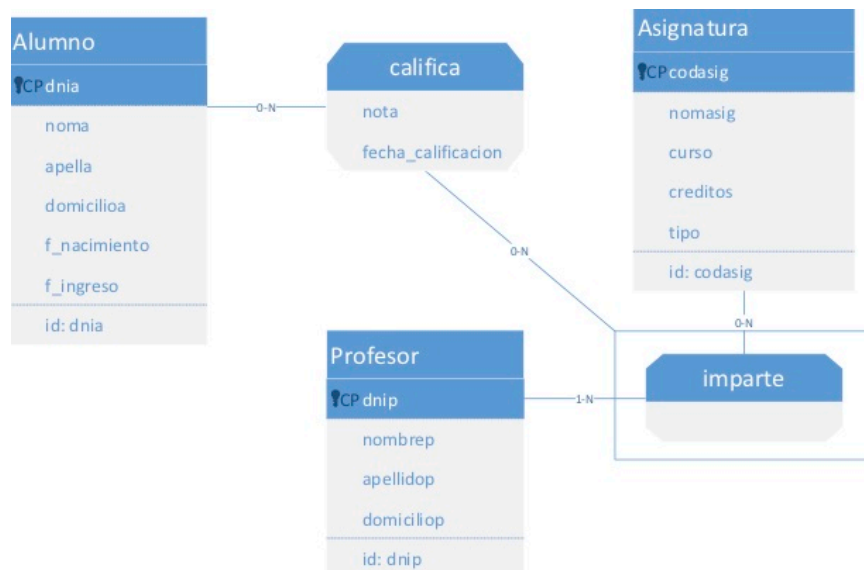
∞ Ejemplo (continuación):

```
CREATE TABLE LINEA
(
    NUMRED decimal(3,0),
    NLINEA decimal(2,0),
    LONGITUD decimal(10,2) NOT NULL,
    CONSTRAINT PK_LINEA PRIMARY KEY (NUMRED, NLINEA),
    CONSTRAINT FK_LINEA_RED_DISTRIBUCION FOREIGN KEY (NUMRED) REFERENCES
        RED_DISTRIBUCION (NUMRED)
);
```

Agregación

- Una **agregación** es una relación entre dos entidades que se trata como si fuera una sola entidad.
 - La agregación es habitualmente una relación de tipo N:N.
 - Se resuelve la agregación y, a continuación, se trata como una entidad que puede estar relacionada con otras entidades.
 - La relación con otras entidades se resolverá en función de su tipo.

Ejemplo:



Agregación

Agregación

∞ Ejemplo (continuación):

```
CREATE TABLE IMPARTE
(
    CODASIG varchar(8),
    DNIP varchar(8),
    CONSTRAINT PK_IMPARTE PRIMARY KEY (CODASIG, DNIP),
    CONSTRAINT FK_IMPARTE_ASIGNATURA FOREIGN KEY (CODASIG) REFERENCES
        ASIGNATURA (CODASIG),
    CONSTRAINT FK_IMPARTE_PROFESOR FOREIGN KEY (DNIP) REFERENCES PROFESOR
        (DNIP)
);
```

Agregación

∞ Ejemplo (continuación):

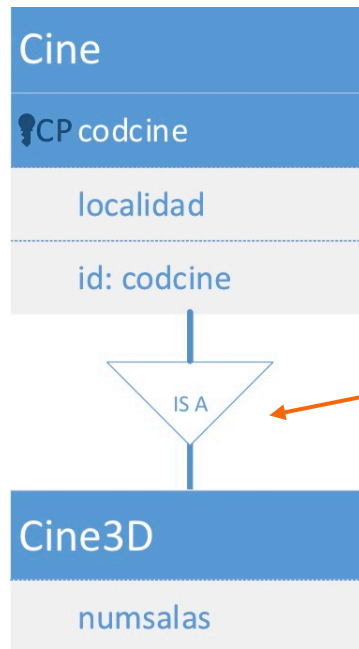
```
CREATE TABLE CALIFICA
(
    DNIA varchar(8),
    CODASIG varchar(8),
    DNIP varchar(8),
    NOTA decimal(2,0),
    FECHA_CALIFICACION date,
    CONSTRAINT PK_CALIFICA PRIMARY KEY (DNIA, CODASIG, DNIP),
    CONSTRAINT FK_CALIFICA_ALUMNO FOREIGN KEY (DNIA) REFERENCES ALUMNO
        (DNIA),
    CONSTRAINT FK_CALIFICA_IMPARTE FOREIGN KEY (CODASIG, DNIP) REFERENCES
        IMPARTE (CODASIG, DNIP)
);
```

Generalización / Especialización

- ∞ **Generalización:** enfoque bottom-up en el que dos (o más) entidades de nivel inferior se combinan para formar una entidad de nivel superior.
- ∞ **Especialización:** enfoque top-down en el que una entidad de nivel superior se divide en dos (o más) entidades de nivel inferior.
- ∞ Las entidades de nivel inferior “heredan” el atributo(s) de la clave primaria de la entidad de nivel superior.
 - Clave primaria: el atributo(s) heredado.
 - Clave externa: el atributo(s) heredado referencia a la tabla de la entidad de nivel superior.

Generalización / Especialización

∞ Ejemplo:



Generalización / Especialización

Generalización / Especialización

∞ Ejemplo (continuación):

```
CREATE TABLE CINE
(
  CODCINE varchar(4),
  LOCALIDAD varchar(20),
  CONSTRAINT PK_CINE PRIMARY KEY (CODCINE)
);
```

```
CREATE TABLE CINE3D
(
  CODCINE varchar(4),
  NUMSALAS decimal(1,0),
  CONSTRAINT PK_CINE3D PRIMARY KEY (CODCINE),
  CONSTRAINT FK_CINE3D_CINE FOREIGN KEY (CODCINE) REFERENCES CINE
    (CODCINE)
);
```

Resumen

1. Recomendación general: no inventar nada que no aparezca en el modelo ER.
 - Si hay líneas (relaciones) hay claves externas, en caso contrario no las hay.
2. Relación 1:N
 - Clave externa en la tabla de la “entidad N” que referencia a la tabla de la “entidad 1”.
3. Relación N:N
 - Nueva tabla cuya clave primaria es, al menos, la combinación de las claves primarias de las entidades participantes. Además, tendrá claves externas que referencian a las tablas de esas entidades.