

Transformación do Esquema Entidade Relación a Esquema Relacional

1.	Transformación do Modelo Conceptual MER ao Modelo Lóxico Relacional de Codd	3
1.1	Principios básicos. O grafo relacional	3
1.2	Transformacións principais baseadas no modelo básico	4
1.3	Regra 1: Transformación de dominios	5
1.4	Regra 2: Transformación de entidades	6
1.5	Regra 3: Transformación de atributos de entidades	6
1.5.1	Regra 3.1: Atributos identificadores principais (AIP)	6
1.5.2	Regra 3.2: Atributos identificadores alternativos (AIA)	6
1.5.3	Regra 3.3: Atributos non identificadores	7
1.6	Regra 4: Transformación de atributos multivaluados	8
1.7	Regra 5: Transformación de atributos derivados	9
1.8	Regra 6: Transformación de interrelacións	11
1.8.1	Regra 6.1. Transformación de interrelacións N:M (monarias e binarias)	11
1.8.2	Regra 6.2. Transformación de interrelacións 1:N (binarias e monaria)	13
1.8.3	Regra 6.3. Transformación de interrelacións 1:1	16
1.8.4	Regra 6.4. Transformación de atributos de interrelacións	20
1.8.5	Regra 6.5. Transformación de interrelación n_area (grado maior que dous)	21
1.9	Regra 7. Transformación de restricións	23

1. Transformación do Modelo Conceptual MER ao Modelo Lóxico Relacional de Codd

1.1 Principios básicos. O grafo relacional

O grafo relacional, tamén denominado grafo de combinación ou diagrama esquemático, é a representación dun sistema mediante un conxunto de relacións vinculadas entre si por unha ou varias claves alleas, empregando unha simboloxía concreta que detallamos a continuación.

É o resultado gráfico da transformación do esquema conceptual entidade-interrelación ao esquema lóxico estándar. A partir del poderase crear a BD coa asistencia do SXBD seleccionado.

Para a súa obtención emprégase unha técnica moi sinxela que permite incluír información sobre:

- Atributos.
- Claves primarias.
- Claves alternativas.
- Claves alleas.

A representación utilizada é a proposta por Adoración de Miguel e Mario Piattini (aínda que existen outras propostas e versións):

- Cada táboa represéntase por un nome de relación e un conxunto de atributos entre paréntese.
- Os atributos que compoñan a clave principal mostraranse subliñados e neste documento engádese a característica de cor vermella.
- Os atributos que compoñan a clave alternativa en estilo de fonte groso.
- Os atributos que compoñan a clave allea visualizaranse en cor vermella e estilo de fonte cursiva, e dende estes debuxaranse uns arcos con punta de frecha que cheguen ata a clave principal ou alternativa á que se fai referencia.
- Os atributos opcionais especifícanse cun asterisco.

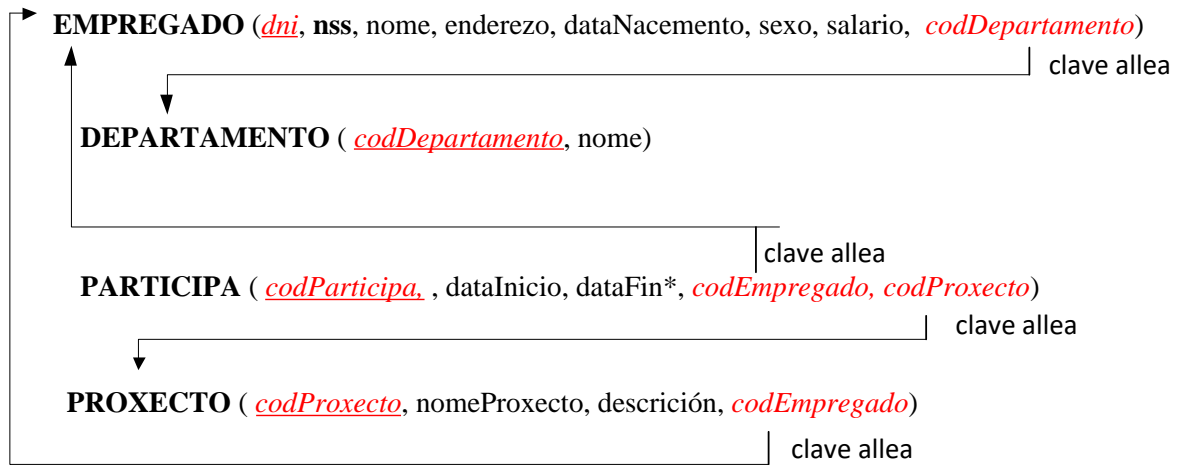


Figura. Exemplo de grafo relacional

1.2 Transformacións principais baseadas no modelo básico

O paso dun esquema no modelo MER de Chen ao relacional de Codd fundaméntase nos tres principios seguintes:

- a) Todo tipo de entidade convértese nunha relación que mantén atributos e claves respecto a entidade de orixe.
- b) Todo tipo de interrelación con tipo de correspondencia N:M, transfórmase nunha relación mediante a técnica de propagación de clave; nela, figurarán como claves alleas as claves principais das relacións procedentes da conversión de tipos de entidade que se relacionaban a través da interrelación.
- c) Todo tipo de interrelación con tipo de correspondencia 1:N, ou ben, se traduce no fenómeno de propagación de clave, ou ben, xera unha nova relación mediante a técnica de propagación de clave.

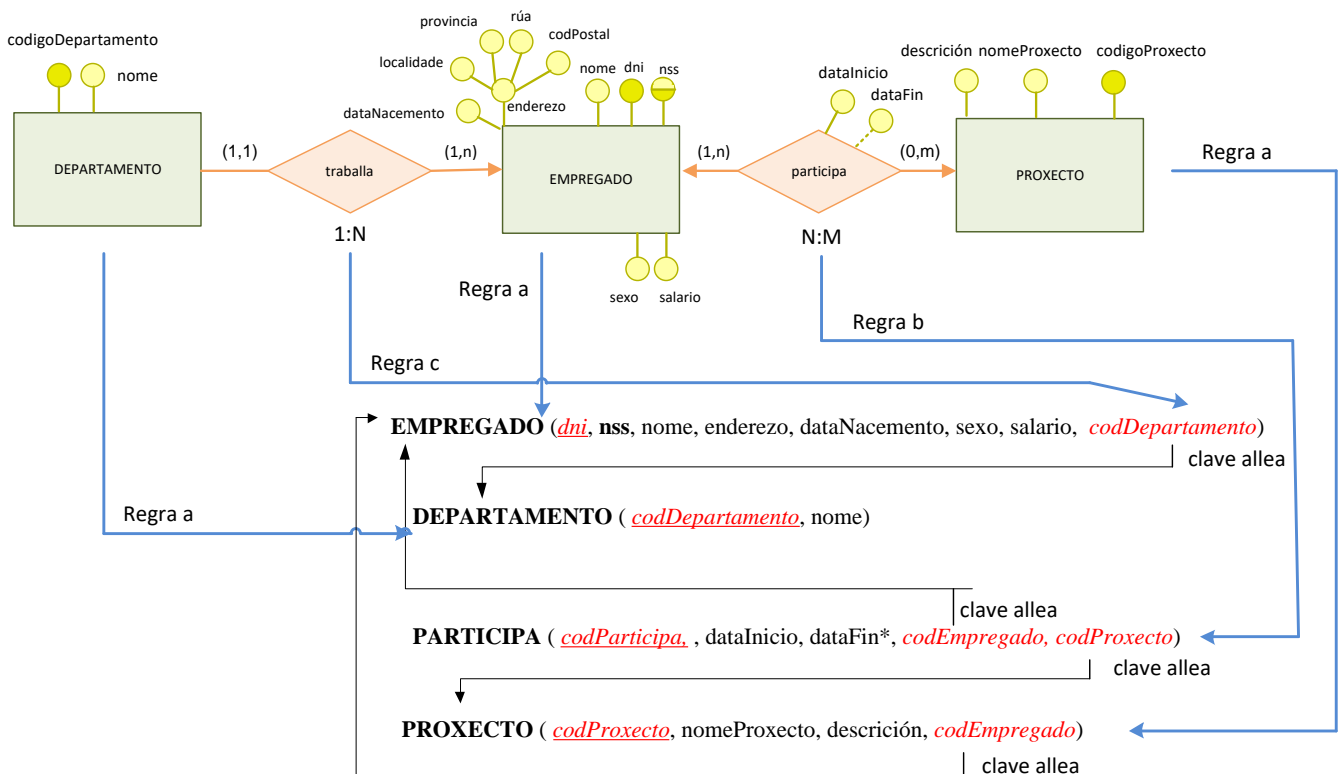


Figura. Exemplo de aplicación das regras básicas de transformación

Aplicando as regras como se pode ver na figura anterior, as táboas resultantes serán EMPREGADO, PROXECTO, DEPARTAMENTO e PARTICIPA, onde a última corresponde á transformación do tipo de interrelación “participa” pola aplicación da regra b. Esta interrelación debe ter como claves alleas as principais de PROXECTO e EMPREGADO. Así mesmo, a interrelación “traballa” tradúcese nunha propagación de clave de DEPARTAMENTO á relación EMPREGADO (a clave principal de DEPARTAMENTO esténdese como clave allea en EMPREGADO).

Na relación PARTICIPA non se aplicou a regra xenérica que indica que as relacións que xorden dunha interrelación N:M deben ter como clave principal a concatenación das claves principais das entidades interrelacionadas; neste caso a razón da conduta radica en evitar unha clave composta de tres atributos (as dúas alleas máis a data de Inicio para que cumpra o principio de unicidade). A razón e evitar unha clave composta é o custo do manexo de índices con claves grandes.

O modelo relacional completo é o que se mostra no grafo relacional. Debido a que este modelo non distingue entre Entidades e interrelacións, ambos conceptos represéntanse mediante relacións. Isto implica unha perda de semántica con respecto o modelo entidade-interrelación:

- As interrelacións N:M non se distinguen das entidades, xa que, ambas transfórmanse en táboas.
- As interrelacións 1:N sóense representar mediante unha propagación de clave desaparecendo o nome da relación, aínda que este nome pode ser asignado ás clave alleas xeradas.

Seguidamente estudarase o modo en que o grafo relacional permite recoller os diferentes elementos, restricións, etc, do modelo relacional. Empregarase o SQL2 xunto con outros elementos como disparadores e procedementos, pendentes da súa estandarización nun futuro a través do SQL3, para representar case na súa totalidade o modelo MER e MERE.

1.3 Regra 1: Transformación de dominios

Os dominios presentes no modelo entidade-interrelación, forman tamén parte do modelo relacional estándar. SQL2 presenta a sentenza CREATE DOMAIN para a súa creación. Así mesmo, é posible engadir restricións á definición relacional de dominio, de forma que este se asemelle máis o concepto de dominio do modelo entidade-interrelación.

Exemplo: Crear o dominio días da semana cos valores de tipo char, de lonxitude 10, e cos valores do rango correspondente aos nomes dos sete días da semana:

```
CREATE DOMAIN diaSemana AS CHAR(10)
CHECK (VALUE IN ('luns', 'martes', 'mercores', 'xoves', 'venres',
'sabado', 'domingo'))
```

O uso de dominios é altamente recomendable, xa que ofrece maior nivel de coherencia, homoxeneidade e robustez ao código resultante.

Ademais de permitir crear dominios, os DDL dos SXBD comerciais soen implementar unha serie de dominios básicos, coñecidos como tipos de datos. Os máis comúns recóllen-se na seguinte táboa:

Dominio	Posibilidades
Numérico	smallInt
	integer/int
	decimal (<i>díxitos significativos, posicións decimais</i>)
	float
	real
Alfanumérico	char (<i>lonxitude cadea</i>)
	var char (<i>lonxitude cadea</i>)
Data (algúns xestores almacenan a hora neste dominio evitando o resto de dominios de hora e data)	Date
Hora	time
Data e hora	Timestamp

1.4 Regra 2: Transformación de entidades

Entre os principios básicos, aparece o que establece que cada tipo de entidade se transforma nunha relación, que se denominará co mesmo nome que a entidade da que provén. Para a súa definición, o DDL-SQL emprega a sentenza CREATE TABLE.

1.5 Regra 3: Transformación de atributos de entidades

Cada atributo dunha entidade transformarase nunha columna da relación creada a partir do tipo de entidade. Tendo en conta que existen atributos identificadores (principal, alternativo) e non identificadores, podemos subdividir esta regra en tres:

1.5.1 Regra 3.1: Atributos identificadores principais (AIP)

Os atributos que son identificadores principais de cada tipo de entidade pasan a formar a clave primaria da relación. Por exemplo, o atributo AIP dni da entidade EMPREGADO xera a clave principal dni na relación EMPREGADO.

Na linguaxe lóxico-estándar do modelo relacional (SQL) emprégase a cláusula PRIMARY KEY para identificar este concepto na descrición da táboa dentro da orde CREATE TABLE. Esta cláusula pode empregarse a nivel de columna, ou, cando o AIP é composto, a nivel de táboa.

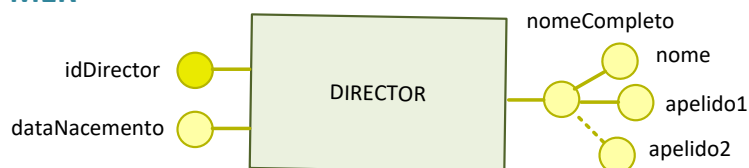
1.5.2 Regra 3.2: Atributos identificadores alternativos (AIA)

Os atributos identificadores alternativos exprésanse en SQL mediante a cláusula UNIQUE na definición de columna ou columnas implicadas dentro da orde CREATE TABLE. Débese forzar se é o caso a que esta clave non tome o valor NULO mediante a sentenza SQL NOT NULL xa que por defecto admítense os NULOS.

1.5.3 Regra 3.3: Atributos non identificadores

Estes atributos forman as columnas da relación á que pertencen, os cales tomarán por defecto a posibilidade de ser NULOS. Para evitalo indícase o contrario mediante a cláusula NOT NULL.

MER



Intensión MR

DIRECTOR (idDirector, nome, apelido1,apelido2*,dataNacemento)

Extensión MR

idDirector	nome	apelido1	apelido2	dataNacemento
987654987	Antón	Reixa	Rodríguez	17-04-1957
236785439	José Luís	Cuerda	Martínez	18-02-1947
879654190	Isabel	Coixet	Castillo	09-04-1960
368965478	Vincenzo	Natali	NULO	06-01-1969
456952134	Steve Allan	Spielberg	NULO	18-12-1946
567423908	Hans Christian	Tomas	Alfredson	01-04-1965

clave primaria

Figura. Exemplo de transformación do tipo de entidade DIRECTOR e os seus atributos

Aplicando as regras vistas ata agora, a transformación dunha entidade ao modelo relacional estándar representarase mediante o seguinte código onde se empregan os dominios correspondentes que han de ser definidos previamente.

```
CREATE TABLE DIRECTOR(  
  idDirector Codigos PRIMARY KEY,  
  Nome Nomes15 NOT NULL,  
  Apelido1 Nomes25 NOT NULL,  
  Apelido2 Nomes25 NULL,  
  dataNacemento datas NOT NULL,  
)
```

1.6 Regra 4: Transformación de atributos multivaluados

Posto que o modelo relacional non permite dominios multivaluados (que nunha mesma celda da táboa exista máis dun valor), deberá crearse unha nova relación composta por ese atributo e a clave primaria da entidade á que pertence como clave allea, da mesma forma que cos tipos de interrelación 1:N. A concatenación de ambos atributos formará a clave primaria da nova relación.



Intensión MR

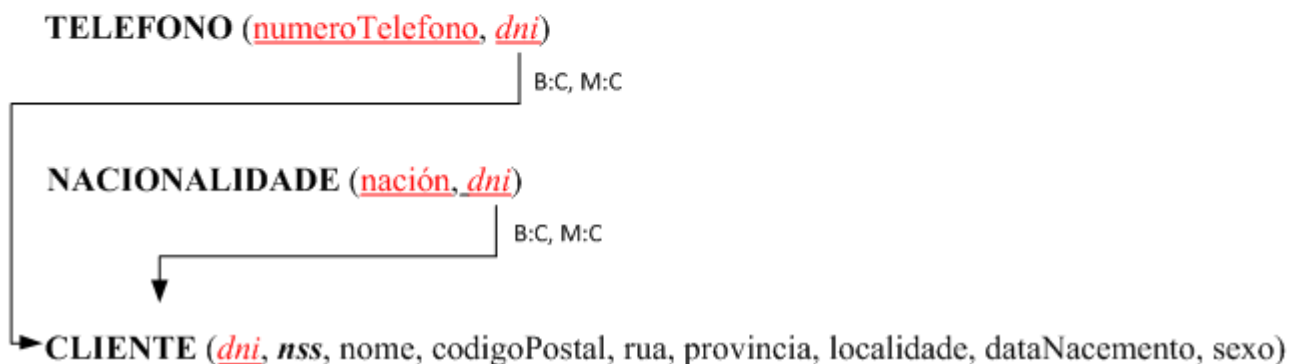


Figura. Exemplo de transformación dun atributo multivaluado

En SQL as claves alleas represéntanse coa cláusula *FOREIGN KEY nomeClaveAllea REFERENCES* dentro da sentenza de creación de táboa. Por cada clave allea obtida deberá estudarse o comportamento da relación que contén a clave allea respecto aos borrados e modificacións de tuplas nas táboas que conteñen as claves candidatas referenciadas (opcións SQL ON DELETE e ON UPDATE). As opcións permitidas en SQL estándar son:

- Operación restrinxida: en caso de non especificar a acción ou NON ACTION.
- Operación de posta a nulo: SET NULL, lembrando que require a opcionalidade dos atributos de clave allea.
- Operación posta a valor por defecto: SET DEFAULT, lembrando que require dar un valor por defecto ao atributo de clave allea ao crear a relación. Esta operación non está recoñecida polo xestor de base de datos MySQL.
- Operación de propagación en cascada: CASCADE


```

CREATE TABLE CLIENTE (
    dni DNIS PRIMARY KEY,
    nss NSS NOT NULL UNIQUE,
    nome Nome6060 NOT NULL,
    codPostal CodigosPostais NOT NULL,
    rua Nomes25 NOT NULL,
    provincia Nomes25 NOT NULL,
    localidade Nomes25 NOT NULL,
    dataNascimento Datas NOT NULL,
    sexo Sexos NOT NULL,
)
CREATE TABLE TELEFONOS (
    numeroTelefono Telefonos,
    dni DNIS,
    FOREIGN KEY (dni) REFERENCES Cliente(dni)
    ON DELETE CASCADE ON UPDATE CASCADE,
    PRIMARY KEY (dni, numeroTelefono),
)
CREATE TABLE NACIONALIDADE (
    nacion Naciones,
    dni DNIS,
    FOREIGN KEY (dni) REFERENCES Cliente(dni)
    ON DELETE CASCADE ON UPDATE CASCADE,
    PRIMARY KEY (dni, nacion),
)

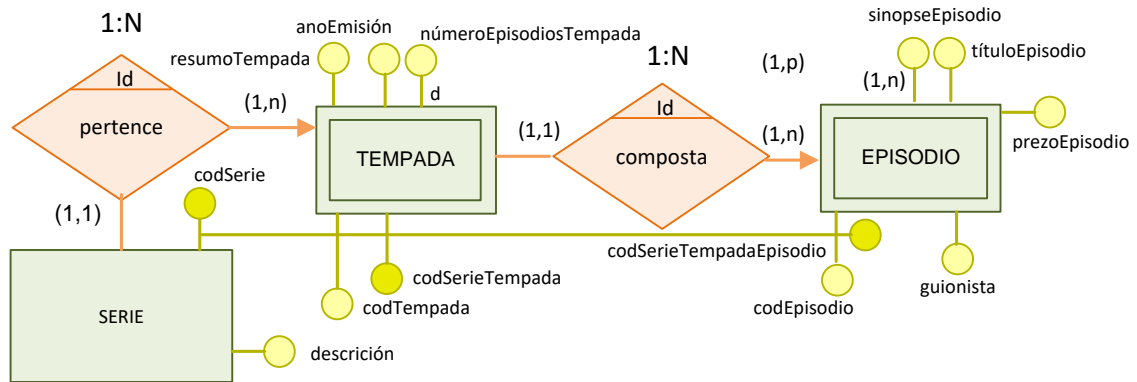
```

1.7 Regra 5: Transformación de atributos derivados

Os atributos derivados son aqueles no que o seu valor se calcula a partir de atributos de entidades ou interrelacións ou do reconto das súas ocorrencias.

Non existe para estes atributos unha representación concreta no modelo relacional, polo tanto, trátanse como atributos normais xerando unha columna da relación á que pertencen. Ademais deberase construír un disparador ou trigger que calcule o valor do atributo cada vez que se engadan ou borren as ocorrencias dos atributos que interveñen no cálculo, e engadir as restricións correspondentes.

MER



Intensión MR

```

graph TD
    SERIE["SERIE ( codSerie, descripción )"]
    TEMPADA["TEMPADA ( codSerie, codTempada, resumenTempada, anoEmision, numEpisodiosTempada )"]
    EPISODIO["EPISODIO ( codSerie, codTempada, codEpisodio, tituloEpisodio, sinopseEpisodio, prezoEpisodio )"]

    SERIE -- "B:C, M:C" --> TEMPADA
    TEMPADA -- "B:C, M:C" --> EPISODIO
    EPISODIO -- "B:C, M:C" --> TEMPADA

```

Figura. Exemplo de transformación do atributo derivado “numeroEpisodiosTempada”

No exemplo, o cálculo do número de episodios dunha tempada sería derivado xa que se calculan a partir do número de ocorrencias da entidade EPISODIO, para cada unha das ocorrencias da entidade TEMPADA. Para manter o atributo actualizado, programárase un disparador para as operacións de alta e baixa das ocorrencias de EPISODIO.

```
CREATE TRIGGER increNumEpisodio AFTER INSERT OF EPISODIO
REFERENCING NEW AS novoEpisodio
FOR EACH ROW
BEGIN
    UPDATE Tempada SET numEpisodiosTempada= numEpisodiosTempada + 1
    WHERE (Tempada.CodSerie == novoEpisodio.CodSerie) AND (Tempada.
CodTempada == novoEpisodio.CodTempada)
END

CREATE TRIGGER decreNumEpisodio AFTER DELETE OF EPISODIO
REFERENCING OLD AS antigoEpisodio
FOR EACH ROW
BEGIN
    UPDATE Tempada SET numEpisodiosTempada= numEpisodiosTempada - 1
    WHERE (Tempada. CodSerie == novoEpisodio.CodSerie)
    AND (Tempada. CodTempada == novoEpisodio.CodTempada)
END
```

1.8 Regra 6: Transformación de interrelacións

O esquema de transformación a seguir dependerá do tipo de correspondencia da interrelación.

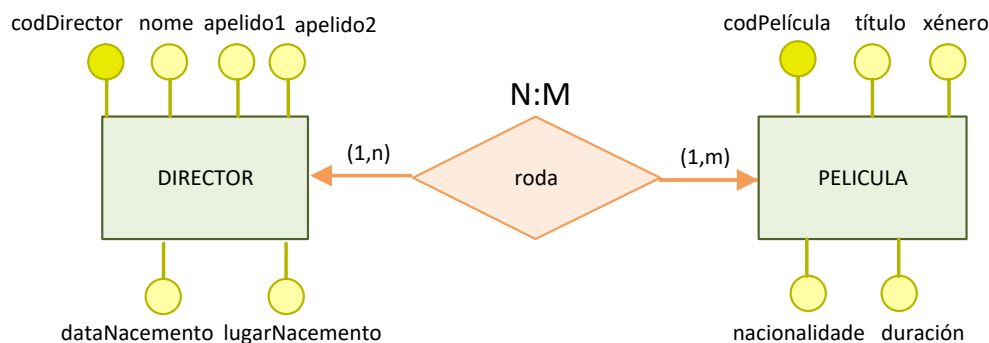
1.8.1 Regra 6.1. Transformación de interrelacións N:M (monarias e binarias)

Un tipo de interrelación N:M transformarase nunha relación ou táboa que terá sendas claves alleas apuntando a cada unha das claves primarias correspondentes aos AIP das entidades tipo que asocia (tamén se poden empregar as claves candidatas para este fin, xa que non existe ningún tipo de impedimento conceptual, aínda que non é o máis habitual).

Esta nova táboa segundo as normas de deseño aconsellase que sexa nomeada co nome da interrelación da que procede, ou coa concatenación dos nomes das entidades que une, para evitar a perda semántica que aparece ao non poder diferenciar táboas procedentes da transformación de entidades e táboas procedentes da transformación de interrelacións. Un exemplo da utilidade deste nomeado son os casos de enxeñería inversa onde non se dispón da documentación do deseño conceptual.

No relativo á clave principal da relación xurdida, o máis habitual é que estea composta pola concatenación das claves primarias das entidades tipo asociadas. Nalgúns casos non chega estes dous atributos para identificar univocamente as tuplas da relación, precisando anexar outro atributo para diferencialas. Noutros casos o tamaño da clave xurdida é tan grande que é preferible a creación dunha clave propia para unha manipulación máis eficiente dos datos.

MER



Intensión MR

PELICULA (codPelícula, título, xenero, nacionalidade, duracion)

↑
B:C, M:C
RODA (codPelícula, codDirector)

↓
B:C, M:C
DIRECTOR (codDirector, nome, apelido1, apelido2, dataNacemento, lugarNacemento)

Figura . Transformación xenérica dunha interrelación binaria N:M

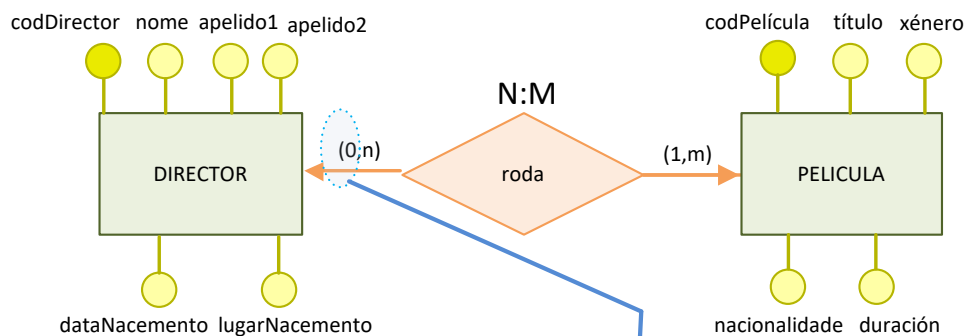
Cardinalidades

Outra característica que debe figurar no MR son algunhas das cardinalidades máxima e/ou mínima das entidades que participan nas interrelacións. No caso das entidades que reciben unha clave allea por propagación, o normal é recoller o valor mínimo 0 (como se amosa no exemplo seguinte) mediante a admisión de nulos (o que obriga a xerar unha clave para a nova relación) e o valor mínimo 1 como a non admisión deste valor nulo.

No relativo aos valores mínimos en clave allea (que non sexan 0 ou 1), para impoñer estas restricións requírese o uso de asertos ou disparadores, aínda que debido a súas limitacións certas normas de deseño non recomendan o seu uso.

Respecto ás cardinalidades máximas, empregan tamén asertos e disparadores aínda que é moito máis inusual a súa implementación.

MER



Intensión MR

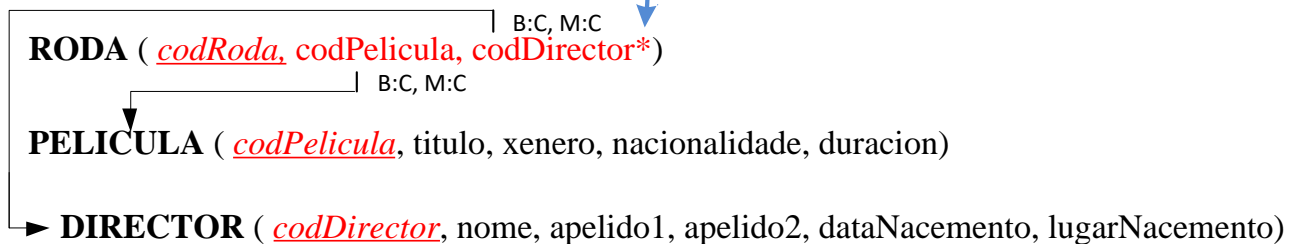


Figura. Transformación da interrelación “roda” con cardinalidade mínima 0 para director

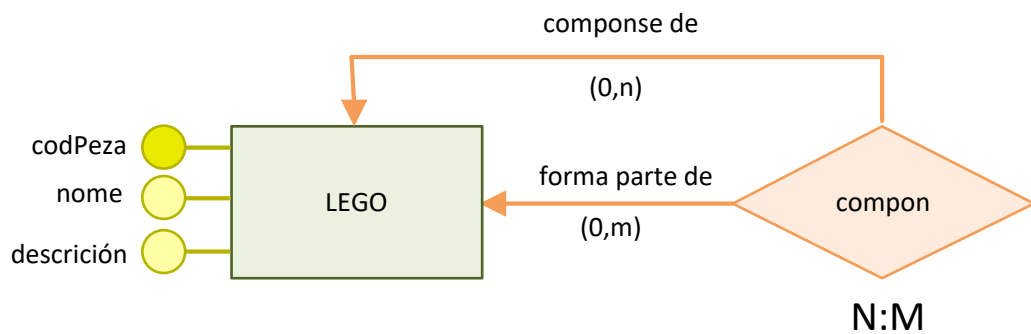
No exemplo anterior, a cardinalidade mínima 0 responde a dúas opcións semánticas:

- Descoñecemos temporalmente o director.
- A película é anónima e nunca se coñecerá quen a rodou.

Para discriminar ambas situacións, decídese que as películas anónimas aparezan na táboa da interrelación polo que agora a clave principal non pode ser a concatenación das claves alleas (xa que sería en parte nula, o que está prohibido polo modelo relacional.), creando unha clave principal nova para a relación “roda” xerada.

Transformación de interrelacións N:M reflexivas ou monarias

O comportamento de transformación é o mesmo que para as interrelacións de tipo correspondencia N:M binarias, tendo en conta que ao migar a clave principal dúas veces, o nome da mesma debe modificarse para que non apareza repetida na nova táboa.



Intensión MR

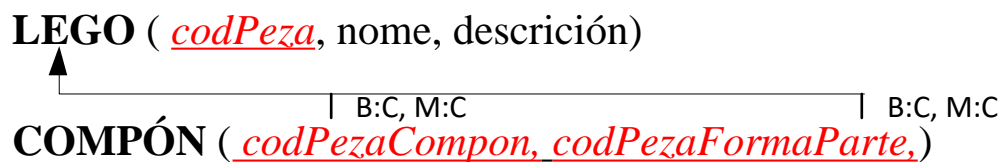


Figura 1.29. Transformación de interrelación reflexiva con tipo de correspondencia N:M

1.8.2 Regra 6.2. Transformación de interrelacións 1:N (binarias e monaria)

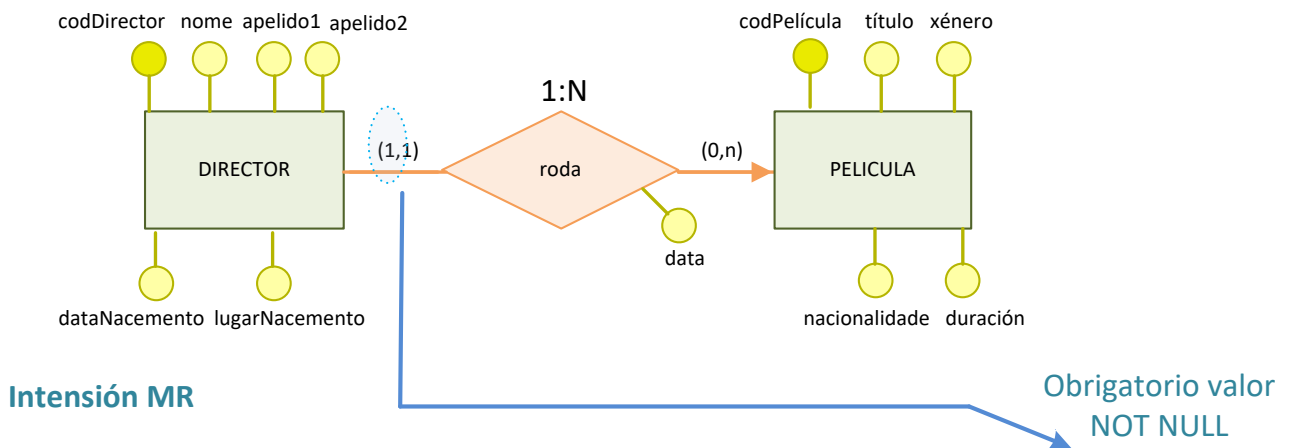
As interrelacións con tipos de correspondencia 1:N pódense transformar utilizando o modelo relacional mediante dous métodos:

- I. Creación dunha restrición de integridade referencial (clave allea) que reflecta a propagación do AIP do tipo de entidade con ocorrencias que se presentan como máximo nunha ocorrencia da relación (o lado 1), ata o tipo de entidade nas que as ocorrencias pódense presentar varias veces (o lado N). A propagación ocorre ata o lado da frecha que é a que simboliza o lado N. Esta é a regra habitual.

Ademais, se a interrelación dispón de atributos propios, estes tamén migran a relación receptora (lado N) sempre e cando non exista perda semántica ou sexa pouco relevante. Se a cardinalidade mínima é de 0, entón os atributos que compoñen a clave allea deben **de permitir** a súa posta a NULO, pero se a cardinalidade mínima é 1 entón os atributos de clave allea **non deben** permitir o valor NULO.

Como se mostra no seguinte exemplo, existe unha perda de semántica no grafo relacional xa que nel se descoñece a procedencia do atributo data, esta perda pode liquidarse mediante co uso de comentarios ou mediante un nomeado especial onde se engada o nome da interrelación para recoñecer os atributos migrados

MER



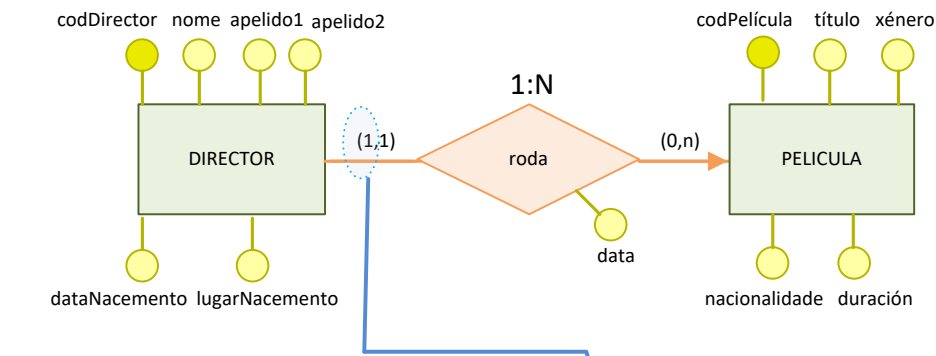
PELICULA (codPelícula, título, xénero, nacionalidade, duración, data, codDirector)
| B:C, M:C

► **DIRECTOR** (codDirector, nome, apelido1, apelido2, , dataNacemento, lugarNacemento)

Figura. Exemplo de transformación dunha interrelación 1:N empregando propagación de claves (solución I)

- II. Aplicar o mesmo tratamento que as relacións N:M, xerando unha nova relación que terá como atributos as claves de ambas entidades xunto cos atributos da interrelación, pero a súa clave será unicamente a da entidade que participa na interrelación con N ocorrencias. Este procedemento resérvase aos seguintes casos:
 - O número de ocorrencias interrelacionadas da entidade que propaga a súa clave é moi pequeno, e polo tanto existirán moitos valores nulos na clave propagada (no exemplo, que moitas películas non teñan un director asociado).
 - Prevese que a interrelación nun futuro converterase nunha de tipo N:M deixando aberta esta posibilidade, xa que a transformación de tipo N:M é mais flexible (no exemplo, unha película sexa rodada por varios directores).
 - Cando a interrelación ten atributos propios non sendo desexable a súa propagación, conservando así a semántica orixinal. Este proceder é o recomendable nas monarias ou reflexivas (no exemplo desexase coñecer certos datos da rodaxe na interrelación).

MER



Intensión MR

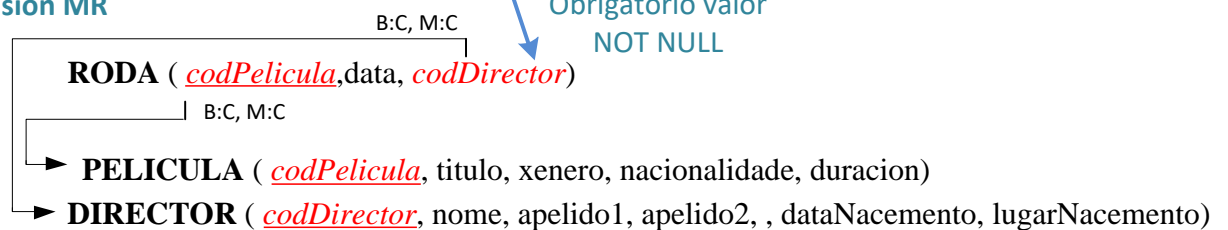


Figura. Exemplo de transformación dunha interrelación 1:N xerando unha nova relación (solución II)

Transformación de interrelación 1:N monarioas ou reflexivas

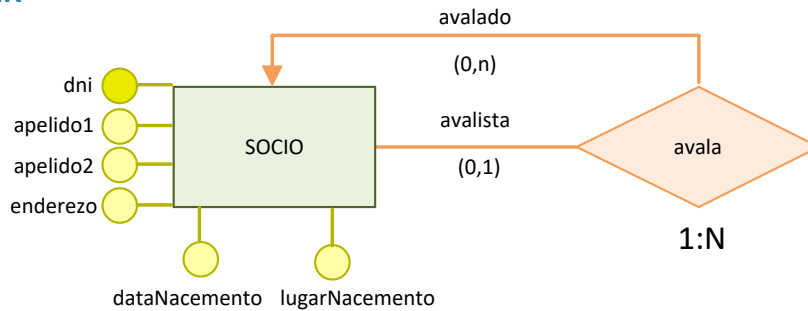
Este tipo de interrelacións reflexivas compórtanse igual que as binarias 1:N, isto é, ou migra a clave ou xera unha nova relación.

No caso de migrar a clave, ao estar na mesma táboa o nome da clave allea debe modificarse para que non apareza repetida, **ademais esta solución non permite os borrados en cascada xa que ao xerar un ciclo podería borrar o contido da relación na súa totalidade.**

Aínda que a norma de carácter xeral que existe nos SXBD cando se definen as claves alleas é crear antes a táboa referenciada que a táboa á que se referencia, estas interrelacións constitúen un caso especial, xa que un tipo de entidade interrelaciónase consigo mesma xerando un bucle. Polo que para crear adecuadamente a clave allea deberíase:

- Definir a táboa mediante CREATE TABLE... sen definir a clave allea pero si os atributos que a compoñen.
- Definir a clave allea engadindo a restrición FOREIGN KEY mediante a sentenza de modificación de táboas ALTER TABLE (nomeTaboaModificar)

MER



Intensión MR

SOCIO (dni, apelido1, apelido2, enderezo, dataNacemento, lugarNacemento, *dniAvalado*)

↑

B:R, M:C

Figura. Exemplo de transformación de interrelación reflexiva con tipo de correspondencia 1:N

No exemplo anterior representase o feito de que un socio concreto non avala a ningún, ou a varios socios, e que un socio pode ser avalado por un ou ningún socio.

1.8.3 Regra 6.3. Transformación de interrelacións 1:1

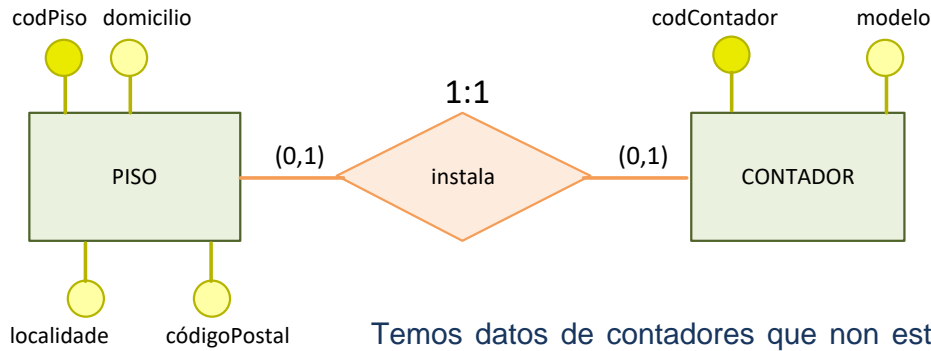
Unha interrelación de tipo de correspondencia 1:1 pódese considerar un caso particular dos tipos de correspondencia 1:N, polo que se poden aplicar as dúas opcións xa comentadas: crear unha nova táboa ou propagación de clave, tendo en conta que agora a propagación de clave pode efectuarse en ambos sentidos. Os criterios para a aplicación dunha regra ou outra, así como para propagar a clave baséanse en:

- Cardinalidades mínimas aplicadas.
- Conservación da semántica do modelo.
- Evitar valores nulos.
- Aumento da eficiencia de procesamento.

A continuación móstrase un exemplo explicando a técnica aplicada:

- Se a interrelación con tipo de correspondencia 1:1 non é obrigatoria para ningunha das entidades participantes, e dicir, se as entidades que se asocian teñen cardinalidades (0,1) e se prevé que a proporción de ocorrencias interrelacionadas non vai ser moi alta ou simplemente non se desexa a aparición de valores nulos na clave allea, entón a interrelación transformarase mediante a **creación dunha nova relación**:

MER



Temos datos de contadores que non están instalados, e de pisos que aínda non teñen contador

Intensión MR

PISO (*codPiso*, domicilio, localidade, códigoPostal)

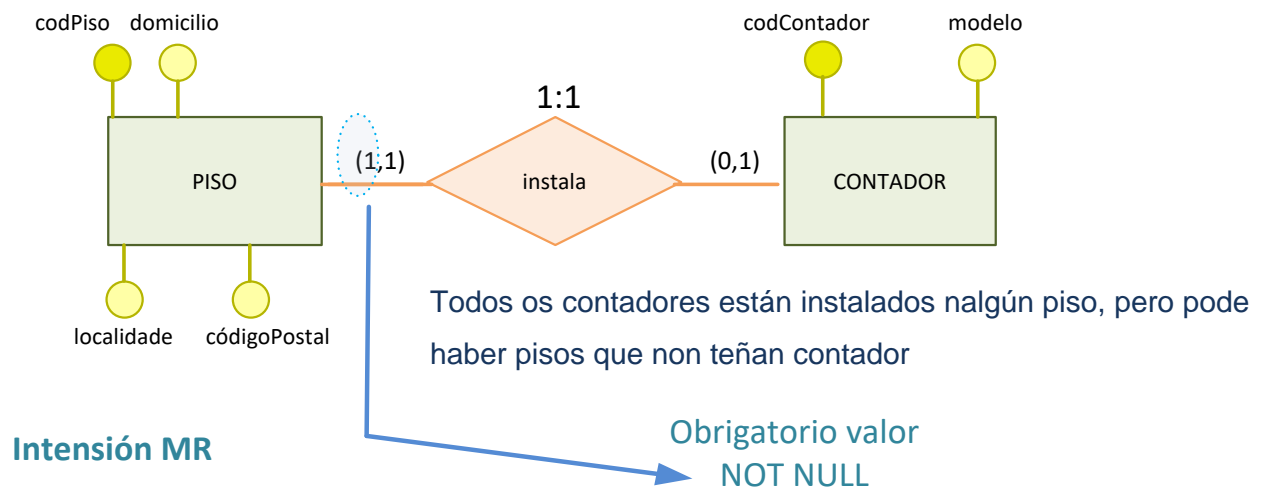
↑
INSTALA (*codContador*, *codPiso*)
 | B:C, M:C

↓
CONTADOR (*codContador*, modelo)
 | B:C, M:C

Figura. Transformación de interrelación 1:1 xerando unha nova relación

- Se a interrelación é obrigatoria para só unha das entidades interrelacionadas, é dicir, cando unha posúe cardinalidades (0,1) e a outra (1,1), e sempre que non se trate dunha relación monaria con atributos, a alternativa é propagar a clave da entidade na que as súas ocorrencias participen obrigatoriamamente na interrelación (propagar da relación resultante da entidade con cardinalidades (1,1) á relación resultante da entidade con cardinalidades (0,1)). Tamén se migraría á mesma relación os atributos da interrelación, o que non debería facerse en ningún caso é a propagación en sentido contrario xa que xeraría nulos.

MER



Intención MR

CONTADOR (codContador, modelo, codPiso)
 B:C, M:C

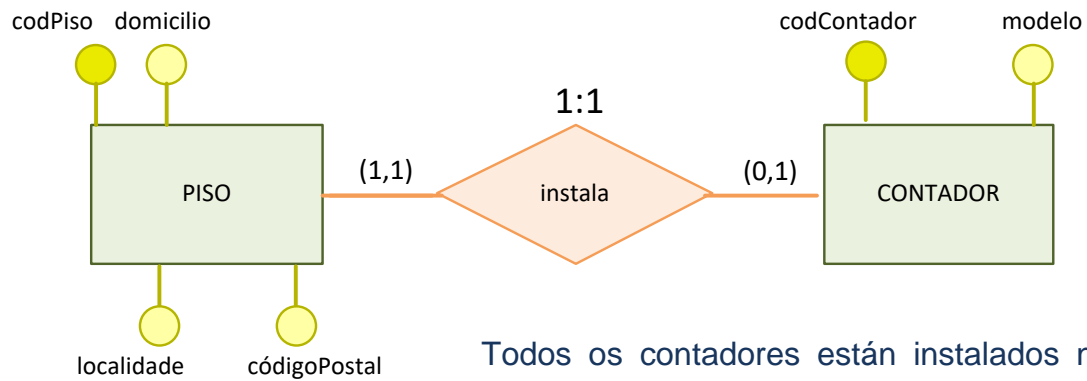
↓

PISO (codPiso, domicilio, localidade, códigoPostal)

Figura. Transformación de interrelación 1:1 propagando clave

- No caso de que a interrelación sexa obrigatoria para todas as entidades interrelacionadas, e dicir, ambas entidades presentan cardinalidades (1,1), poderase propagar a clave de calquera delas á relación resultante da outra; tendo en conta neste caso que é mellor que reciba a clave aquela relación que vaia ser accedida máis frecuentemente. **Outra posibilidade é priorizar a eficiencia propagando a clave en ambos sentidos. Esta solución produce redundancia a controlar por restricións adicionais, polo que as normas de deseño NON aconsellan esta opción.**

MER



Todos os contadores están instalados nalgún piso, pero pode haber pisos que non teñan contador

Intensión MR

PISO (*codPiso*, domicilio, localidade, códigoPostal)



CONTADOR (*codContador*, modelo, *codPiso*)

| B:C, M:C

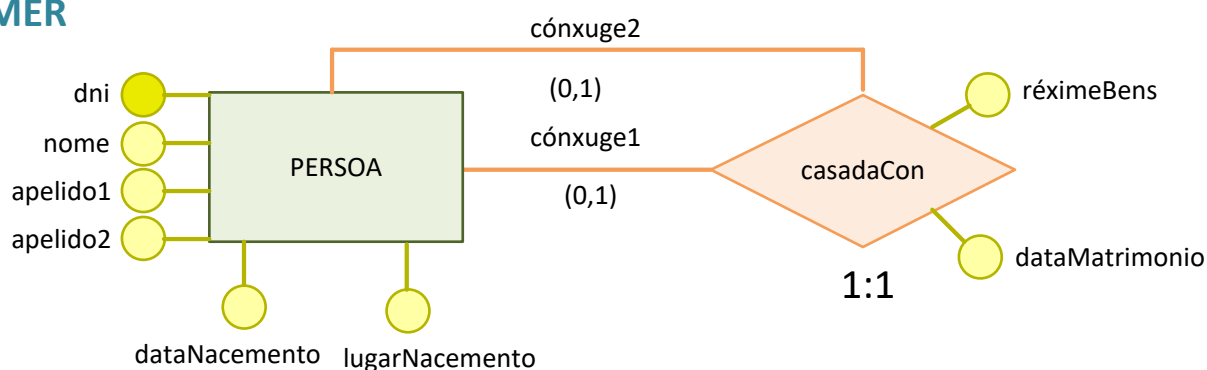
Figura. Transformación de interrelación 1:1 supoñendo un maior acceso aos datos contidos na entidade CONTADOR

Monarias ou reflexivas con tipo de correspondencia 1:1

Con este tipo de interrelación actuarase preferentemente creando unha táboa, preferencia que se converterá en obrigatoriedade cando a relación monaria sexa portadora de atributos.

Aínda que é posible a propagación de clave sobre a mesma táboa, debe recordarse que esta solución non permite os borrados en cascada.

MER



Intensión MR

PERSOA (*dni*, nome, apelido1, apelido2, dataNascimento, lugarNascimento)

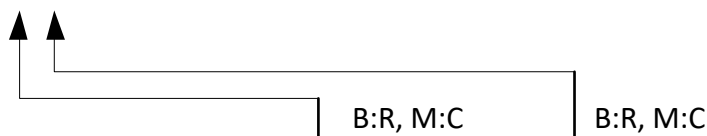
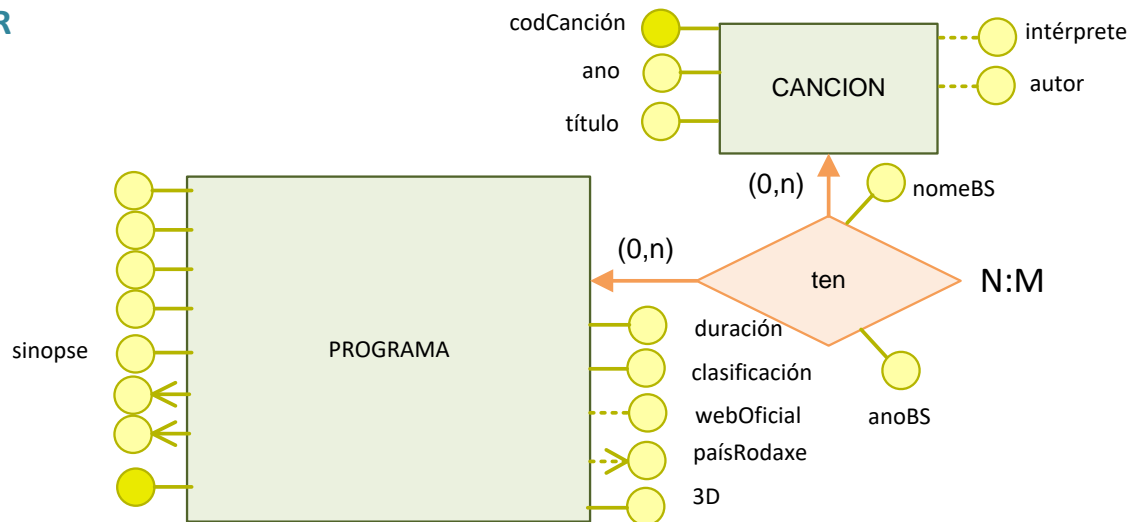
**MATRIMONIO** (*dniConxuge1*, *dniConxuge2*, dataMatrimonio, reximeBens)

Figura. Transformación de interrelación reflexiva con tipo de correspondencia 1:1

1.8.4 Regra 6.4.Transformación de atributos de interrelações

Nos apartados anteriores viuse que se unha interrelación se transforma nunha relación, todos os seus atributos pasan a ser atributos desta nova relación, mentres que cando se representa mediante a propagación de clave, os seus atributos migrarán xunto coa clave á relación que recibe a propagación desta.

MER



Intensión MR

► **CANCION** (*codCancion*, *titulo*, *ano*, *autor**, *interprete**)

BANDASONORA (*CodBandaSonara*, *nomeBS*, *anoBS*, *codPrograma*, *codCancion*)

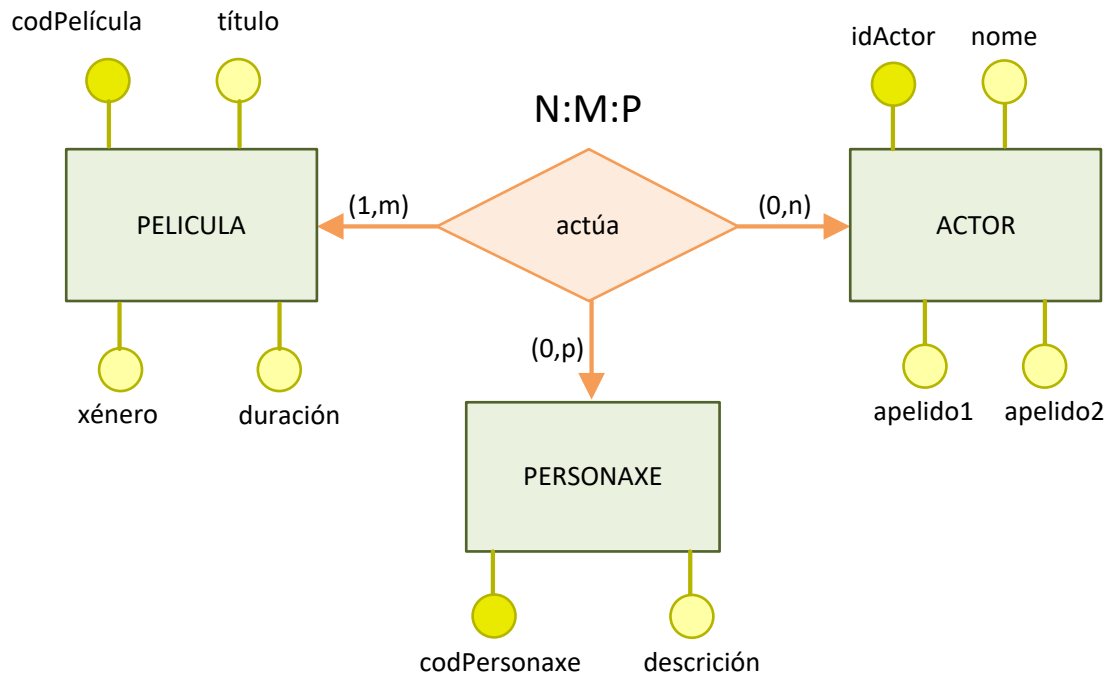
► **PROGRAMA** (*codPrograma*, *idiomaVisionado*, *idiomaSubtitulos*, *tituloDistribucion*, *tituloOrixinal*, *dataEstreo*, *idomaOrixinal*, *sinopse*, *duracion*, *clasificacion*, *webOficial**, *paisRodaxe*, *3D*)

Figura. Exemplo de transformación dunha Interrelación que xera unha nova táboa

1.8.5 Regra 6.5. Transformación de interrelación n_area (grado maior que dous)

As interrelacións ternarias, cuaternarias, etc. represéntanse do mesmo modo que as interrelacións N:M, é dicir, xerando unha nove relación que terá sendas claves alleas apuntando a cada unha das claves correspondentes ás entidades tipo que asocia.

MER



Intensión MR: solución 1

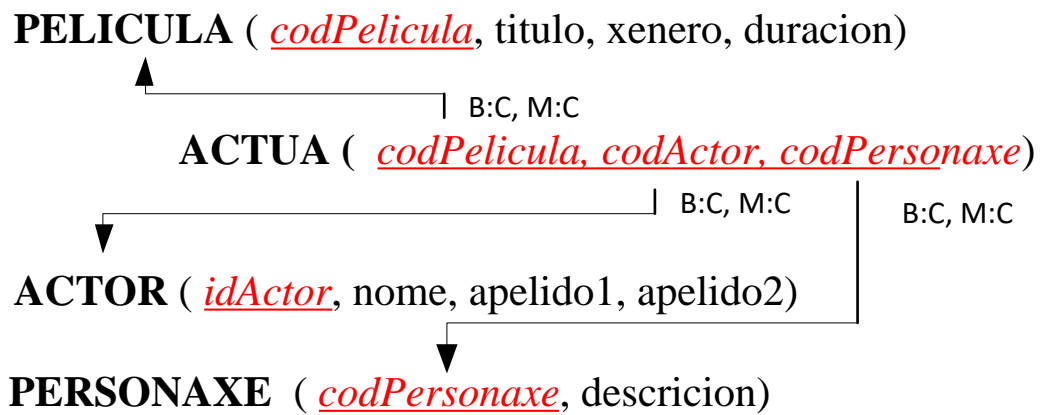


Figura. Exemplo de transformación dunha interrelación ternaria

No tocante a clave principal da relación xurdida, o máis común é que esté formada pola concatenación das claves tipo asociadas.

EXEMPLOS E SITUACIÓNS

Por motivos de eficiencia na xestión interna das claves, optase por crear un atributo clave principal específico ao que se asociará un tipo autoincremental que manexa automaticamente aos SXBD.

Intensión MR: solución 2

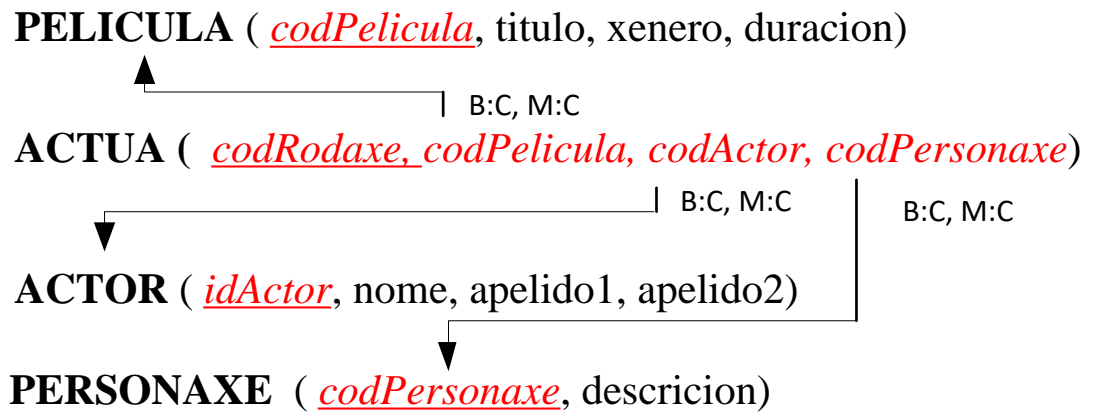


Figura. Transformación de interrelación ternaria para a que se crea a clave “codRodaxe” na relación xerada

Ata o momento explicouse como controlar certos casos de cardinalidade, pero para o total control das cardinalidades mínimas e máximas de cada entidade participante; tendo en conta que as posibilidades increméntanse sustancialmente en comparación coas binarias, deberá recurrirse á implementación de restricións empregando validacións, disparadores, asercións, etc.

1.9 Regra 7. Transformación de restricións

A representación das restricións, tanto as impostas no modelo (integridade referencial, etc) como as semánticas derivadas dos requisitos de usuario (valores positivo para idade, etc), pódense definir mediante as seguintes cláusulas:

- Restricións de dominio: CREATE DOMAIN, NOT NULL, DEFAULT...
- Restricións de integridade referencial: FOREIGN KEY e cláusulas de comportamento asociado (ON DELETE CASCADE, RETRICT...).
- Restricións de comprobación: CHECK (comprobación de condicións de atributos da mesma táboa), BETWEEN (rangos de valores de dominios), IN (enumeración de valores de atributos).
- Asercións: CREATE ASSERTION (permite comprobar condicións entre atributos de diferentes táboas).
- Outras restricións :
 - Disparadores: CREATE TRIGGER.
 - Procedementos: CREATE PROCEDURE.