Modelo relacional

3.1 Introdución

As bases do modelo relacional foron postuladas en 1970 por Edgar Frank Codd, nos laboratorios IBM en San José (California). O modelo relacional non tardou en consolidarse como un novo paradigma nos modelos de base de datos substituíndo aos clásicos sistemas de ficheiros que mostraban grandes deficiencias.

A súa idea fundamental é o uso de **relacións**, constituíndo estas o elemento fundamental deste modelo de aí o seu nome, "**modelo relacional**". Estas relacións poden considerarse como conxuntos de datos chamados **tuplas**. Pese a que esta é a teoría das bases de datos relacionais creadas por **Codd**, a maioría das veces se conceptualiza dunha maneira máis fácil de imaxinar, pensando en cada relación como se fose unha **táboa** que está composta por **rexistros** (cada fila da táboa será un rexistro ou "tupla") e **columnas** (tamén chamadas "campos").

O modelo relacional, para o modelado e xestión de bases de datos, é un modelo baseado na **lóxica de predicados de primeiro orde** e na **teoría de conxuntos**, polo que presenta unha sólida base formal.

Este modelo de datos publicado por Codd perseguía os seguintes **obxectivos**:

- **Independencia física**. O modo no que se almacenan os datos non debe influír na súa manipulación lóxica e, por tanto, os usuarios que acceden a eses datos non han de modificar os seus programas por cambios no almacenamento físico.
- **Independencia lóxica**. Engadir, eliminar ou modificar calquera elemento da BD non debe repercutir nos programas e/ou usuarios que están a acceder a subconxuntos parciais dos mesmos (vistas).
- **Flexibilidade**. Ofrecer a cada usuario os datos da forma máis adecuada á correspondente aplicación.
- **Uniformidade**. As estruturas lóxicas dos datos presentan un aspecto uniforme (táboas), o que facilita a concepción e manipulación da BD por parte dos usuarios.
- **Sinxeleza**. As características anteriores, así como unhas linguaxes de usuario moi sinxelas, producen como resultado que o modelo relacional (MR) sexa fácil de comprender e de utilizar por parte do usuario final.

Codd concedeu moita importancia ao tema da independencia da representación lóxica dos datos respecto do seu almacenamento interno, que concretou en tres tipos de independencia: de ordenación, de indexación, e dos camiños de acceso.

Importancia que Codd manifesta explicitamente:

"... proponse un modelo relacional de datos como unha base para protexer aos usuarios de sistemas de datos formateados dos cambios que potencialmente poden alterar a representación dos datos, causados polo crecemento do banco de datos e polos cambios nos camiños de acceso“.

Os avances máis importantes que o modelo de datos relacional incorpora respecto dos modelos de datos anteriores son:

- **Sinxeleza e uniformidade**. Os usuarios ven a base de datos relacional como unha colección de táboas, e ao ser a táboa a estrutura fundamental do modelo, este goza dunha gran uniformidade, o que unido a unhas linguaxes non navegacionais e moi orientadas ao usuario final, dá como resultado a sinxeleza dos sistemas relacionais.
- **Sólida fundamentación teórica**. Ao estar o modelo definido con rigor matemático, o deseño e a avaliación do mesmo pode realizarse por métodos sistemáticos baseados en abstraccións.
- **Independencia da interface de usuario**. As linguaxes relacionais, ao manipular conxuntos de rexistros, proporcionan unha gran independencia respecto da forma na que os datos están almacenados.

As vantaxes citadas contribuíron a que desde mediados dos anos 80, o MR sexa utilizado por practicamente a totalidade dos SXBD comerciais, sendo o modelo de datos relacional máis utilizado para modelar sistemas reais de aplicacións comerciais de procesamento de datos convencionais. Impúxose debido ás limitacións dos modelos anteriores como o de Codasyl (modelo en rede) ou o xerárquico. Baséase no uso de relacións ou táboas que agrupan conxuntos de datos en forma de filas ou tuplas. Ademais, incorpora **restricións** que son formas de reflectir as regras de funcionamento da empresa ou organización que se está a modelar.

Algunhas das principais empresas informáticas do mundo son en orixe empresas de SXBD (ORACLE, Sybase, INFORMIX, …), os grandes fabricantes de software teñen “o seu” SXBD relacional (IBM DB2, Microsoft SQL Server, ...), ademais existen bastantes SXBD deseñados para PC’s e usuarios non expertos (Microsoft Access...).

3.2 Elementos e terminoloxía

Para construír e interpretar os modelos relacionais necesitamos en primeiro lugar coñecer os seus elementos, que nos permitirán representar os datos do sistema a modelar.

3.2.1 Relación

A relación é o elemento básico do modelo relacional e constitúe unha estrutura de datos cun nome e un conxunto de atributos ou características.

Por exemplo, unha película dunha BD de venda de visionado de programas quedaría representado do seguinte modo:

PROGRAMA (*codPrograma* , *idiomasVisionado, idiomasSubtitulos,* tituloDistribucion, tituloOrixinal, dataEstreo, idomaOrixinal, sinopse, duracion, clasificacion, webOficial*, paisRodaxe, xenero, 3D, tipoPrograma)

Representamos a relación PROGRAMA cos seus campos ou características. Na BD almacenarase os atributos de PROGRAMA xunto co tipo de dato de cada un; especificando o tamaño e o dominio dos mesmos, se son obrigatorios ou opcionais, etc.

3.2.2 Atributo

Os atributos permítenos definir as propiedades ou características da relación.

Por exemplo, a relación PROGRAMA disporá entre outros dos atributos "tituloDistribucion" que representa o título co cal o programa se estreou no noso país ou "webOficial" que conterá a URL do programa.

3.2.2.1 Dominio

O dominio é o conxunto de valores permitidos para certos atributos. Por exemplo, o atributo "xénero" na relación PROGRAMA só poderá tomar os valores do dominio (drama, comedia, biografía, misterio, terror, romance, guerra, animación e aventuras).

Podemos definir un dominio coma un conxunto nomeado (identifícase ou caracterízase por un nome), cun número finito de elementos atómicos ou indivisibles e homoxéneo (do mesmo tipo).

Cada **dominio** pode definirse de dúas maneiras:

- **Extensión** (dando os seus posibles valores).
  Por exemplo, clasificación de programa = { todos os públicos, maiores de 9 anos, maiores de 15 anos, maiores de 18 anos}.
- **Intensión** (mediante un tipo de datos).
  Por exemplo, duración dun programa = enteiro, ou duración dun programa = entre 15 e 300 minutos e título = ata 50 carácteres alfanuméricos. Ás veces asóciase unha unidade de medida (quilos, metros, etc.) e/ou certas restricións (como un rango de valores).

Existe un valor especial non asociado a un dominio concreto que pode permitirse ou non nun dato para reflectir situacións do mundo real onde existe: información perdida, ausencia de información ou valores non aplicables a ese atributo. Para representar este valor os sistemas de bases de datos identifícano como **NULO** (NULL). Se unha tupla ten un valor nulo significa que o valor real dese atributo é descoñecido.

NULO non é un valor en si mesmo, senón un indicador ou marca de ausencia de información. Non hai dous nulos iguais polo que non se pode comparar dous atributos coa marca NULO.

3.2.2.2 Dominio e atributo

Un atributo e un dominio poden chamarse igual, pero os atributos diferéncianse dos dominios:

- Un atributo está sempre asociado a unha relación, mentres que un dominio ten existencia propia con independencia das relacións.
- Un atributo representa unha propiedade dunha relación.
- Un atributo toma valores definidos nun dominio, estando cada atributo definido sobre un único dominio obrigatoriamente.
- Un dominio pode conter valores non tomados por ningún atributo.
- Varios atributos distintos (da mesma ou de diferentes relacións) poden tomar os seus valores do mesmo dominio.

En ocasións, pode acontecer que o valor dun atributo para unha fila sexa descoñecido, neses casos, asóciase a esa ocorrencia un valor NULO para ese atributo. Non debe confundirse un valor nulo cunha cadea de caracteres baleira ou cun valor numérico igual a cero.

A comparación de dous atributos só terá sentido se ambos toman valores do mesmo dominio. Se o SXBD soporta Dominios, poderá detectar este tipo de erros.

Ademais dos dominios e atributos simples, que acabamos de definir, en ampliacións posteriores do MR introduciuse o concepto de dominio composto, que é moi útil na práctica. Un **dominio composto** pódese definir como unha combinación de dominios simples á que se poden aplicar certas restricións de integridade. Por exemplo, o dominio composto denominado Data constrúese por agregación dos dominios simples Día, Mes e Ano, incorporando as adecuadas restricións de integridade a fin de que non aparezan valores non válidos para a data.

Do mesmo xeito que é posible definir dominios compostos, existen tamén atributos compostos. Tanto os **atributos compostos** como os dominios compostos poden ser tratados, se así o precisa o usuario, como “elementos únicos” de información, é dicir, como valores atómicos.

Facendo unha analoxía entre dominios e tipos de datos nas linguaxes de programación; o dominio sería o tipo de datos da linguaxe de programación e cada atributo a variable.

3.3 Tupla

É o conxunto válido de valores que toma cada un dos atributos, son as chamadas **ocorrencias** ou **exemplares** da relación. Por exemplo unha tupla da relación PROGRAMA podería ser:

PROGRAMA (*P00045624*, {español, galego, inglés, francés}, {español, galego, inglés, francés}, The godfather, El padrino, 1972, inglés, 'Michael Corleone é un home apartado dos negocios mafiosos da súa familia ata que a vida do seu pai Don Vito Corleone corre perigo', 175, maiores de 18 anos, NULO, Estados Unidos, drama, NON, película)

3.3.1.1 Cardinalidade

O número de tuplas ou ocorrencias dunha relación denomínase cardinalidade. Así, por exemplo, na relación DIRECTOR da base de datos de venda de visionado teremos unha cardinalidade igual ao número de directores rexistrados (neste caso 6).

DIRECTOR (*987654987*, Antón, Reixa, Rodríguez, 17-04-1957,3 , 00076
236785439, José Luís, Cuerda, Martínez, 18-02-1947, 8, 00076
879654190, Isabel, Coixet, Castillo, 09-04-1960, 3, 00076
368965478, Vincenzo, Natali, NULO, 06-01-1969, 1, 00034
456952134, Steve Allan, Spielberg, NULO, 18-12-1946, 00023
567423908, Hans Christian, Tomas, Alfredson, 01-04-1965, 00012)

3.3.1.2 Grao

O número de atributos dunha relación constituirá o grao. Así, na relación DIRECTOR do exemplo, o grao e igual a 7.

3.4 Restricións

Cando facemos un modelo de datos dun sistema, ademais dos datos e as súas relacións establecemos as restricións, que permiten reflectir certas condicións ou **regras de negocio** que desexamos que cumpran.

Por exemplo, na BD venda de visionado de programas, unha restrición pode ser que cada película teña un código asociado para distinguilas das series. Estas restricións impoñen condicións ao modelo, e poden ser polas características do mesmo ou polos requirimentos do deseñador.

3.4.1 Restricións inherentes ou implícitas

As restricións inherentes derívanse da mesma estrutura do modelo, non tendo que ser definidas polo deseñador xa que é o modelo quen as impón. Estas restricións actívanse no momento da definición do esquema cando se produce un intento de violación do mesmo, rexeitándose todo esquema que non as cumpra.

Ademais das restricións derivadas da definición de relación:

- Cada **relación** ten un **nome distinto**.
- **Non** hai **dous atributos** que se **chamen** igual.

O MR define as seguintes:

- **Non existen tuplas repetidas** (do mesmo xeito que no concepto matemático de conxunto). É dicir, non pode haber dúas filas ou ocorrencias dunha relación co mesmo valor en todos os campos. Esta condición ven imposta pola obrigatoriedade da clave primaria.
- **A orde das tupas é irrelevante**, non requiríndose unha orde determinada nas filas que forman a táboa.
- **A orde dos atributos** (columnas) non é significativa.
- **Cada atributo só pode tomar un único valor do dominio subxacente**, non admitíndose polo tanto os grupos repetitivos (nin outro tipo de estruturas). É dicir, non podemos, por exemplo, ter dous valores para o campo títuloOrixinal na relación de película.

3.4.1.1 Restrición de integridade de entidade

Debe cumprirse a regra de que "**ningún atributo que forme parte da clave primaria dunha relación pode tomar un valor descoñecido ou inexistente (nulo)**".

A regra aplícase sobre as relacións do esquema e só á clave primaria (non ás claves alternativas).

3.4.2 Restricións semánticas

As restricións de integridade semánticas ou de usuario son facilidades que o modelo ofrece aos deseñadores para que poidan reflectir no esquema, o máis fielmente posible, a semántica do mundo real (UD), son impostas polo deseño en función dos requisitos do sistema a modelar e polo tanto dependen do mesmo.

Os tipos de restricións semánticas permitidos no MR (incorporados no estándar SQL 92), permiten aumentar a súa capacidade expresiva, e son:

3.4.2.1 Restricións de integridade

Estas regras de integridade son recoñecidas polo MR e defínense no esquema da base de datos, sendo restricións de comportamento que se activan no momento da actualización da BD, e rexeitan todo exemplar que non as cumpra ou poñen en marcha mecanismos a fin de que non se produza un estado inconsistente (evita configuracións de datos imposibles). As restricións de integridade deben cumprirse pola base de datos en todos os seus estados, formando parte das cabeceiras das relacións.

3.4.2.1.1 Restricións sobre atributos

Podemos distinguir entre:

- **Restricións de dominio**. Nestas restricións ao definir cada atributo sobre un dominio, imponse unha restrición sobre o conxunto de valores permitidos para cada atributo.
- **De obrigatoriedade (valor non nulo)**. Permiten declarar se un ou varios atributos deben tomar sempre un valor, e dicir, non poden tomar valores descoñecidos ou nulos.

3.4.2.1.2 Restrición de unicidade (UNIQUE RESTRICTION)

Permite definir claves candidatas, prohibindo que os valores dun ou varios atributos se repitan en diferentes tuplas. Unha relación pode ter varias **claves candidatas**, entre as que se seleccionará unha **clave de identificación primaria**. As non seleccionadas denominaranse **claves alternativas**.

Clave Candidata (CC, CANDIDATE KEY –CK-) e clave alternativa (CAl, ALTERNATIVE KEY –AK-)

A clave é o conxunto de atributos que identifican univocamente cada tupla nunha relación (non existen dúas tuplas distintas co mesmo valor para a clave) e minimamente (ningún subconxunto de atributos da clave ten a propiedade de unicidade).

A clave candidata pode ser:

- **Simple**. Formada por un só atributo.
- **Composta**. Formada por máis dun atributo.

  CLIENTE (*dni*, **numeroTarxeta**, apelido1, apelido2, rua, localidade, provincia, codigopostal, correoElectronico*, telefono, dataNacemento, idade)

Para o caso da relación CLIENTE, as posibles claves candidatas serán o dni e o número de tarxeta. Porén, a unión do dni e o nome non sería unha clave candidata mínima xa que non se necesita o nome para identificar un cliente (é polo tanto reducible), de aí a definición de mínimo na clave.

3.4.2.1.3 Restrición de Clave Primaria (PRIMARY KEY RESTRICTION)

Esta restrición permite declarar un atributo ou un conxunto de atributos como clave primaria dunha relación.

Clave Primaria (CP, PRIMARY KEY – PK-)

A clave primaria é a que o deseñador escolle de entre as claves candidatas por motivos alleos ao modelo, cumprindo sempre a regra de integridade de entidade.

Os seus valores non se poderán repetir nin se admitirán os nulos (ou valores “ausentes”).

Aínda que o modelo teórico impón esta restrición, nin SQL92 nin os SXBD's relacionais obrigan á declaración dunha clave primaria para cada táboa, pero permiten a definición da mesma.

No caso da relación CLIENTE, mostrada no exemplo anterior, poderíase escoller o atributo dni como clave primaria deixando o número de tarxeta como clave alternativa.

Debemos distinguir entre a restrición inherente de obrigatoriedade da clave primaria e a restrición semántica que lle permite ao usuario indicar que atributos forman parte da clave primaria.

3.4.2.1.4 Restrición de Integridade Referencial (FOREIGN KEY RESTRICTION)

Esta restrición permítenos unir relacións (ou unha relación consigo mesma) a través dos valores dos atributos, de xeito que o contido dun atributo ou conxunto de atributos nunha relación debe coincidir cos valores da clave primaria da outra relación.

Supoñamos a seguinte representación dunha base de datos sobre un casting previo a un concurso de actores:

ACTOR		
codActor	**nome**	**idade**
456	Iria	18
678	Ximena	21
123	Breixo	32

MEMBRO XURADO		
codXurado	**nome**	**especialidade**
222	Sabela	NULO
333	Roi	NULO
444	Xulia	NULO

PAPEL		
codPapel	**descricion**	**duracion**
1	rapaza	20
2	rapaz	17
3	malo	7
4	amiga	3

ACTUACIÓN		
codPapel	**codActor**	**data**
1	456	03/03/2015
4	456	03/03/2015
1	678	04/03/2015
2	123	04/03/2015

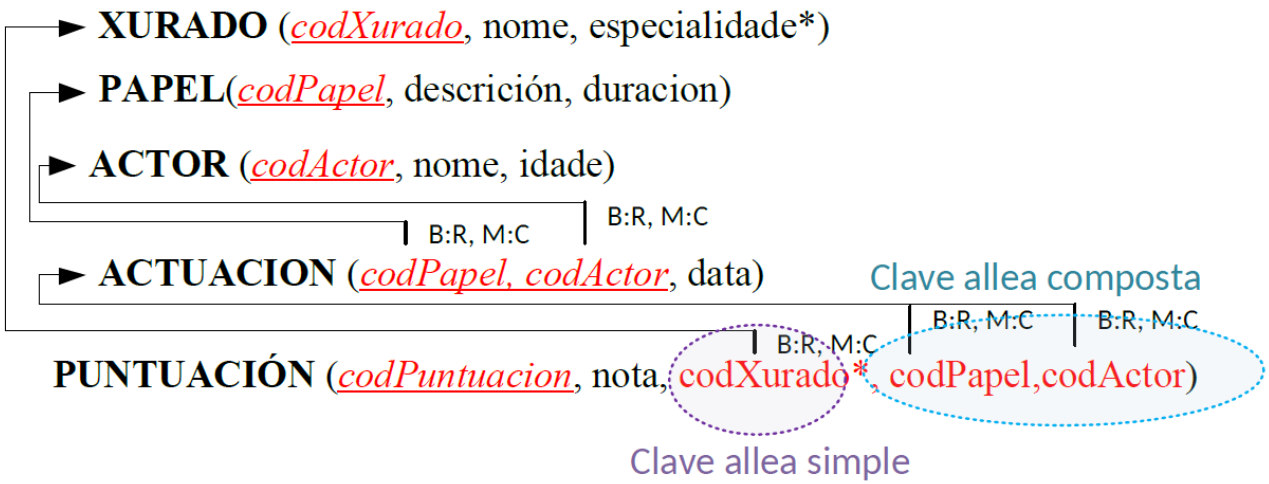
PUNTUACIÓN				
codPuntuacion	**nota**	**codPapel**	**codActor**	**codXurado**
1	3	1	678	222
2	5	4	456	333
3	7	NULO	456	333
4	6	2	123	444
5	8	1	678	222
6	6	3	678	444
7	5	NULO	NULO	333

Clave Allea (foránea ou externa, CA, FOREING KEY –FK-)

Un descritor, atributo ou conxunto de atributos é clave foránea (FK) dunha relación "R2" (relación que referencia) se todo valor de dito descritor concorda cun valor da clave primaria

doutra relación “R1” (relación referenciada). O modelo impide que existan na base de datos valores de claves alleas sen correspondencia cunha clave principal.

A clave allea pode ser: **simple** (formada por un só atributo) ou **composta** (integrada por varios atributos).



Os atributos que son clave allea nunha relación non precisan ter o mesmo nome que o atributo clave primaria coa que se corresponde.

Cada compoñente dunha clave allea (FK) debe estar definido sobre o mesmo dominio que o correspondente atributo da clave primaria (PK) referenciada.

O uso de claves alleas facilitará a eliminación das redundancias, e será o mecanismo do modelo relacional para establecer vínculos entre relacións, dando lugar á estrutura da base de datos.

NULOS e claves alleas:

O modelo relacional permite NULO como valor de clave allea, por exemplo, empregados non asignados a un departamento ou empregados sen xefe. Tendo en conta isto, podemos estender a definición de clave allea:

Un descritor, atributo ou conxunto de atributos é clave foránea (FK) dunha relación “R2” (relación que referencia) se todo valor de dito descritor concorda cun valor da clave primaria doutra relación “R1” (relación referenciada) ou é NULO.

PUNTUACIÓN (*codPuntuacion*, nota, codXurado *, codPapel, codActor)

PUNTUACIÓN				
codPuntuacion	**nota**	**codXurado**	**codPapel**	**codActor**
1	3	222	1	678
2	5	333	4	456
3	7	333	NULO	456
4	6	444	2	123
5	8	222	1	678
6	6	NULO	3	678
7	5	333	NULO	NULO

A regra de Integridade Referencial tendo en conta os NULOS establece que a base de datos non debe conter valores NON NULOS de clave allea sen correspondencia. **PUNTUACIÓN** (*codPuntuacion*, nota, codXurado *, codPapel, codActor)

PUNTUACIÓN				
codPuntuacion	**nota**	**codXurado**	**codPapel**	**codActor**
1	3	222	1	678
2	5	333	4	456
3	7	333	NULO	456
4	6	444	2	123
5	8	222	1	678
6	6	NULO	3	678
7	5	333	NULO	NULO

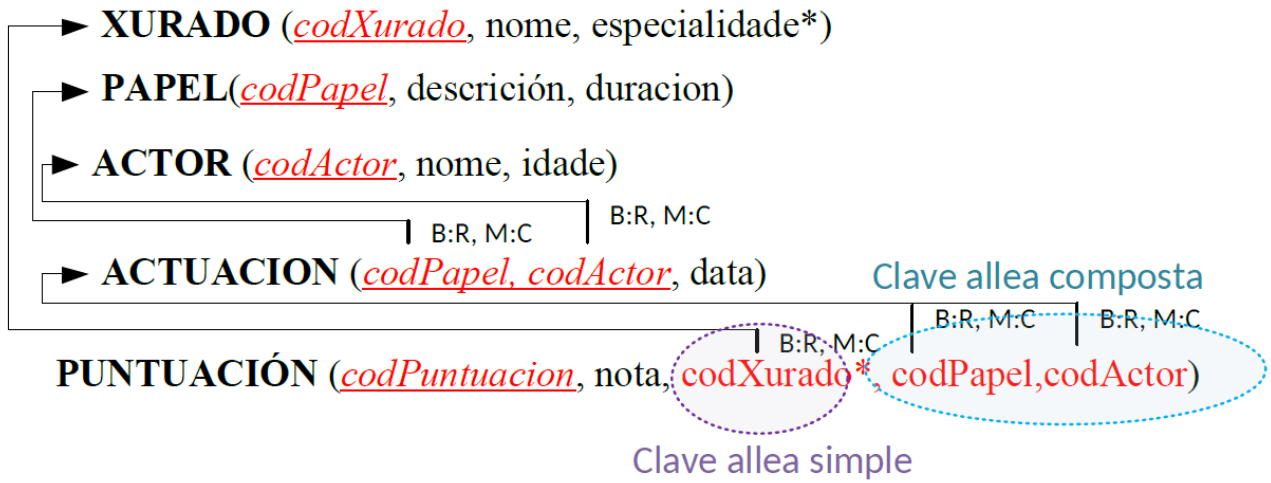
Políticas de mantemento da integridade referencial

O **modelo relacional** permite definir as opcións de **borrado** e **modificación** nas **claves alleas**.

Hai que ter presente que a BD é dinámica e que, polo tanto, os valores vanse modificando e eliminando ao longo do tempo de modo que si creamos restricións de integridade referencial debemos determinar certas accións que deben tomarse como consecuencia destas operacións a realizar sobre as filas das táboas referenciadas. Para evitar estados ilegais o sistema poderá realizar:

- **Rexeitar toda operación que produza un estado ilegal da base de datos**. Este proceso denomínase **operación restrinxida** (RESTRICT): só se pode borrar ou modificar tuplas na relación que ten clave primaria referenciada se non existen filas con esa clave allea na táboa que referencia.
- **Aceptar e executar as operacións** pero realizando accións que restauren a **integridade dos datos**, podendo afectar a dúas relacións ou máis se á súa vez estas relacións están unidas a outras, producíndose cambios en cadea (debe facerse fincapé que se se rexeita unha operación o resto das operacións sobre as relacións afectadas suspéndense.):
  - **Operación con transmisión en cascada (CASCADE)**. O borrado ou a modificación dunha tupla da relación que contén a clave primaria referenciada conleva o borrado ou a modificación en cascada das tuplas da relación que referencia onde a clave allea coincide co valor da clave primaria da táboa referenciada.

- **Operación con posta a nulos (SET NULL)**. O borrado ou a modificación dunha tupla da relación que contén a clave primaria referenciada leva consigo a posta a nulos dos valores da clave allea das tuplas da relación que referencia onde a clave allea coincide co valor da clave primaria da táboa referenciada.
- **Operación con posta a valor por defecto (SET DEFAULT)**. O borrado ou a modificación dunha tupla da relación que contén a clave primaria referenciada leva consigo a posta a un valor por defecto dos valores da clave allea das tuplas da relación que referencia onde a clave allea coincide co valor da clave primaria da táboa referenciada.



A representación gráfica das opcións de borrado utilizada é:

- B:R. Borrado restrinxido.
- B:C. Borrado en cascada.
- B:N. Borrado con posta a nulos.
- B:D. Borrado a un valor por defecto.

A representación gráfica das opcións de modificación utilizada é:

- M:R. Modificación restrinxida.
- M:C. Modificación en cascada.
- M:N. Modificación con posta a nulos.
- M:D. Modificación a un valor por defecto.

As operacións de borrado e modificación poden ser distintas para unha determinada clave allea dunha relación; por exemplo, é posible definir a modificación en cascada e o borrado restrinxido.

3.4.3 Restricións explícitas, baseadas no esquema ou de negocio

Este grupo de restricións son específicas de cada base de datos, sendo definidas e almacenadas no esquema da BD mediante o uso de SQL ou a linguaxe propia do SXBD. Non poden ser violadas por ningunha operación de actualización.

- **Restricións de verificación (CHECK)**. As restricións de verificación tamén chamadas check, permiten impor condicións a elementos ou atributos dunha relación. Por exemplo, podemos impor que o campo idade na relación CLIENTE estea comprendida entre 18 e 100 anos. Comproba en toda operación de actualización, se o predicado é certo ou falso e, no segundo caso, rexeita a operación. A restrición de verificación defínese sobre un único elemento (dentro dun CREATE TABLE) e pode ou non ter nome.
- **Restricións de aserción (ASSERTION)**. Estas restricións son similares ás de verificación, a única diferenza é que agora as condicións poden ser de atributos de máis dunha táboa. Por tanto, a súa definición non vai unida a un determinado elemento do esquema e sempre ha de ter un nome. Por exemplo, podemos impor que un cliente para opinar dunha serie teña que ver ao menos tres episodios da mesma, outro exemplo sería que os empregados que participan nun proxecto sexan do mesmo departamento que o seu responsable.
- **Restrición de disparadores (TRIGGERS)**. Restrición na que o usuario pode especificar libremente a resposta (acción) ante unha determinada condición. Son obxectos programados polo usuario para tal fin. Así como as anteriores regras de integridade son declarativas, os disparadores son procedimentais, sendo preciso que o usuario escriba o procedemento que ha de aplicarse no caso de que se cumpra a condición. Por exemplo, facer que cando unha película se dese de alta o atributo numProgramasDirixe na relación DIRECTOR se incremente nunha unidade na fila correspondente a ese DIRECTOR.

3.5 Transformación do Modelo Conceptual MER ao Modelo Lóxico Relacional de Codd

3.5.1 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 se detalla a continuación.

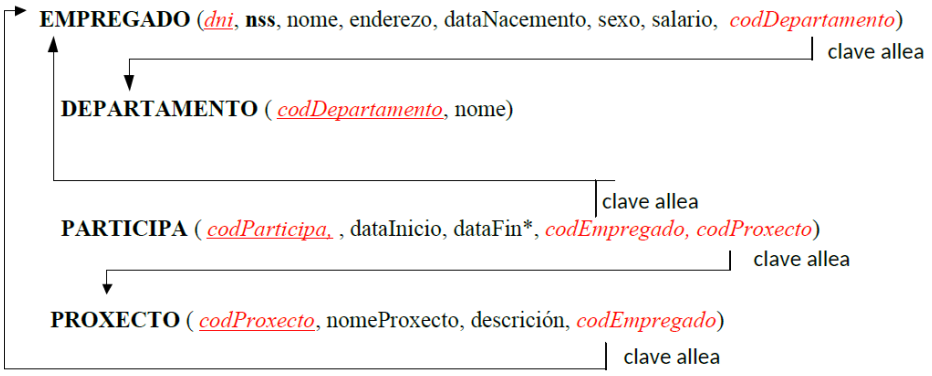
O grafo relacional é o resultado gráfico da transformación do esquema conceptual entidade-interrelación ao esquema lóxico estándar. A partir del poderase crear a BD coa asistencia do SXBD seleccionado.

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.



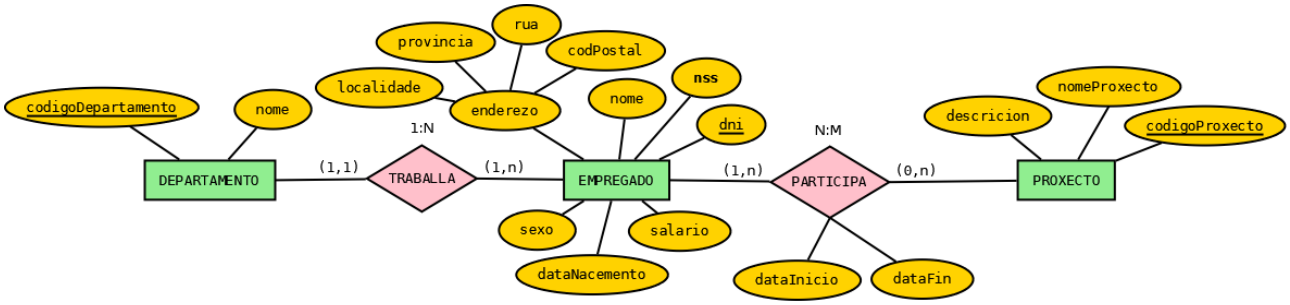
3.5.2 Transformacións principais baseadas no modelo básico

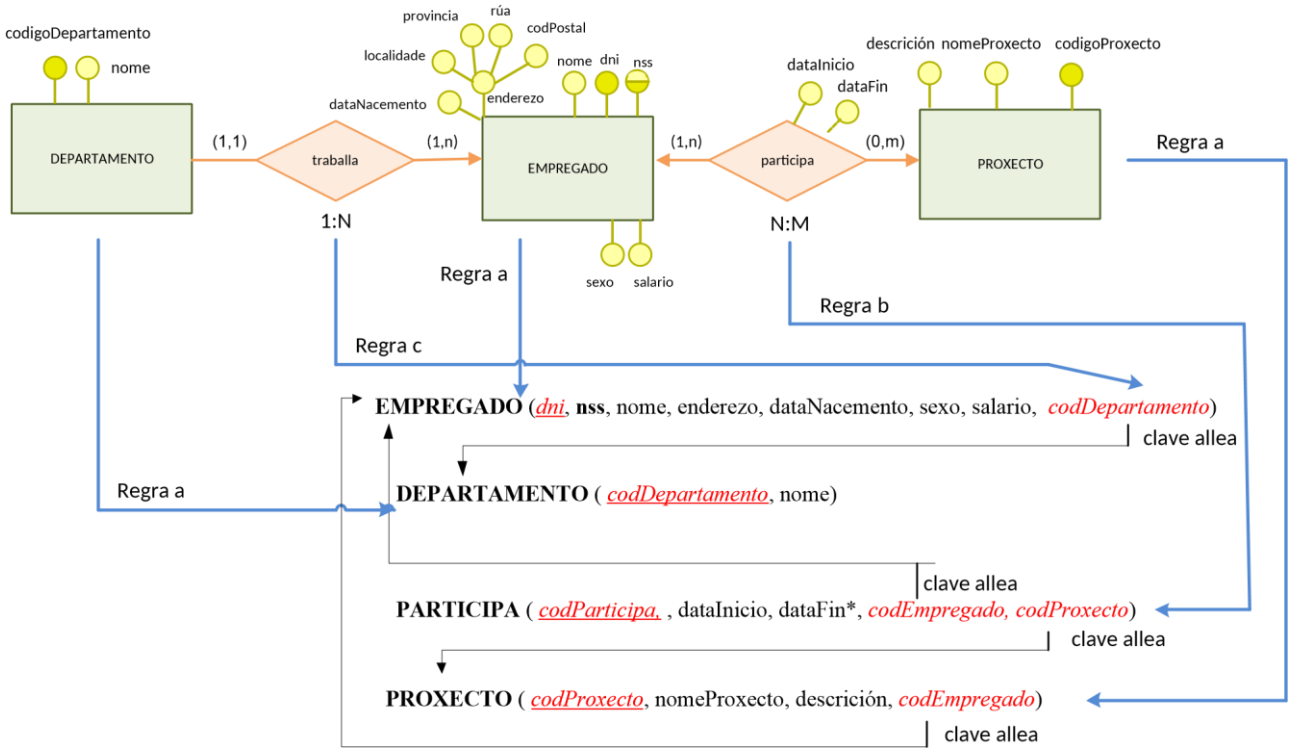
O paso dun esquema no modelo MER de Chen ao relacional de Codd fundaméntase nos tres principios seguintes:

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**.





Aplicando as regras á 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 DEPARTAMENTO. 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 é evitar unha clave composta e o costo 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:

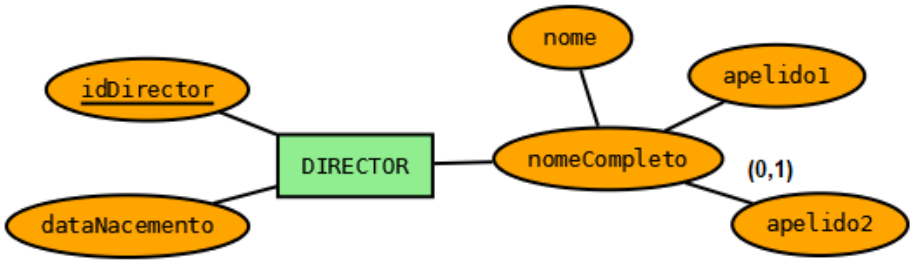
- As interrelacións N:M non se distinguen das entidades, xa que, ambas se transforman en táboas.
- As interrelacións 1:N sóense representar mediante unha propagación de clave desaparecendo o nome da relación, aínda que este nome pode ser asignado ás clave alleas xeradas.

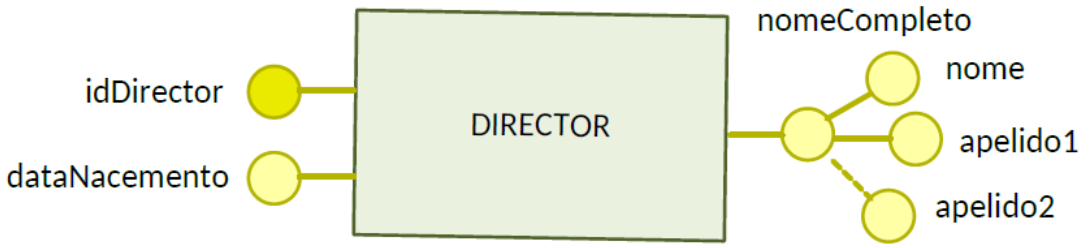
3.5.2.1 Transformación de atributos de entidades

3.5.2.1.1 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

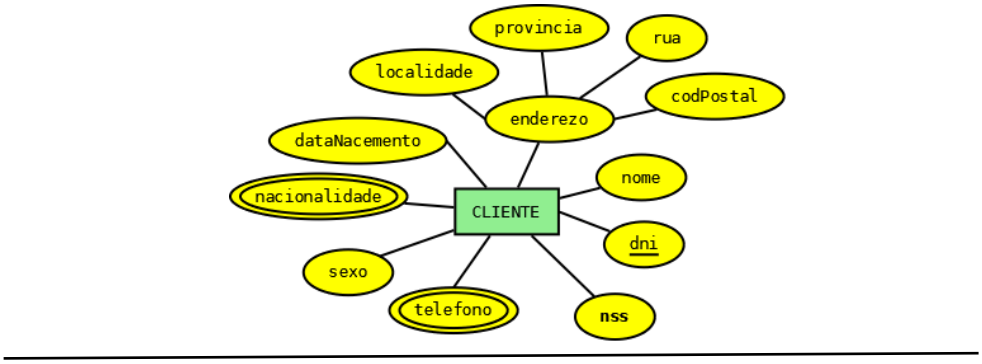
clave primaria

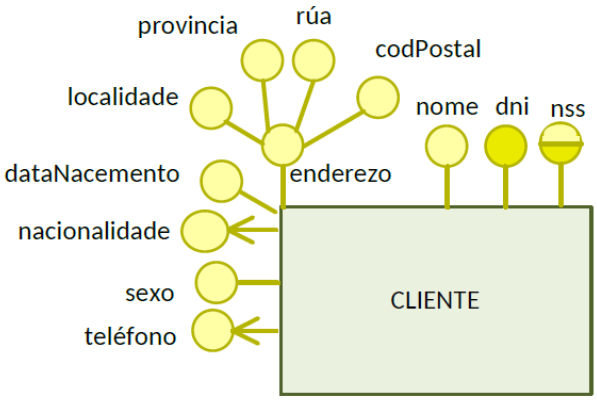
Os **atributos compostos** transfórmanse nos atributos simples (campos) que compoñen o atributo composto, desaparecendo o atributo composto como tal da relación.

3.5.2.1.2 Transformación de atributos multivaluados

Posto que o modelo relacional non permite dominios multivaluados (que nunha mesma cela 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.

MER





Intensión MR

TELEFONO (numeroTelefono, *dni*)

B:C, M:C

NACIONALIDADE(nacion, *dni*)

B:C, M:C

CLIENTE (*dni*, ***nss***, nome, codigoPostal, rua, provincia, localidade, dataNacemento, sexo)

3.5.2.1.3 Transformación de atributos derivados

Os atributos derivados son aqueles nos 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.

Teoricamente, calquera atributo derivado é redundante, pero poden engadirse á base de datos se se consultan con frecuencia.

3.5.2.2 Transformación de interrelacións

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

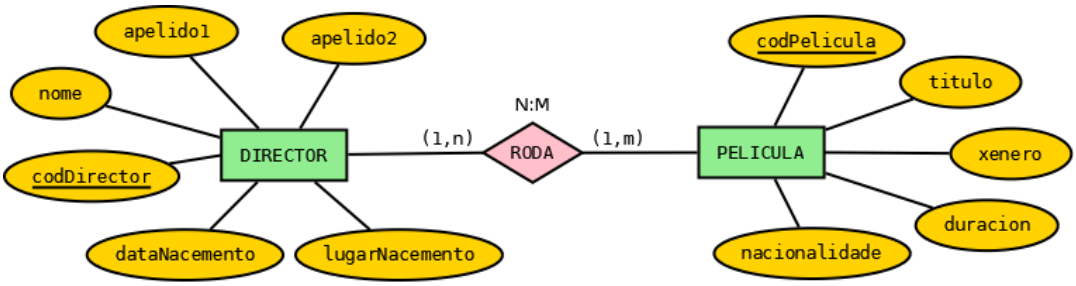
3.5.2.2.1 Transformación de interrelacións N:M

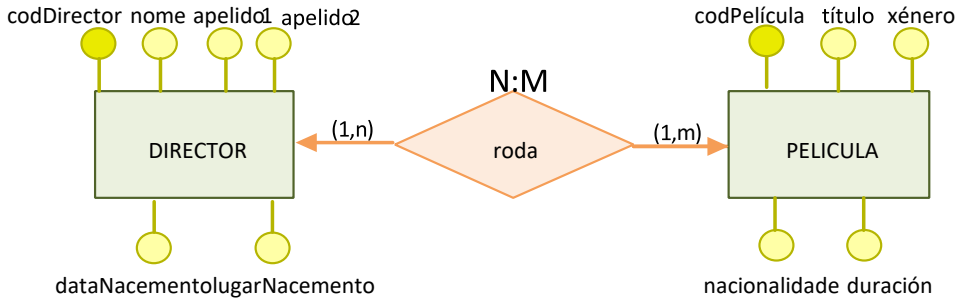
Un tipo de interrelación N:M transformarase nunha relación ou táboa que terá sendas claves alleas apuntando a cada unha das claves primarias correspondentes aos AIP das entidades tipo que asocia (tamén se poden empregar as claves candidatas para este fin, xa que non existe ningún tipo de impedimento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

Esta nova táboa segundo as normas de deseño aconsella 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 máis eficiente manipulación dos datos.

MER





Intensión MR

PELICULA (*codPelicula*, titulo, xenero, nacionalidade, duracion)

B:C, M:C

RODA (*codPelicula*, *codDirector*)

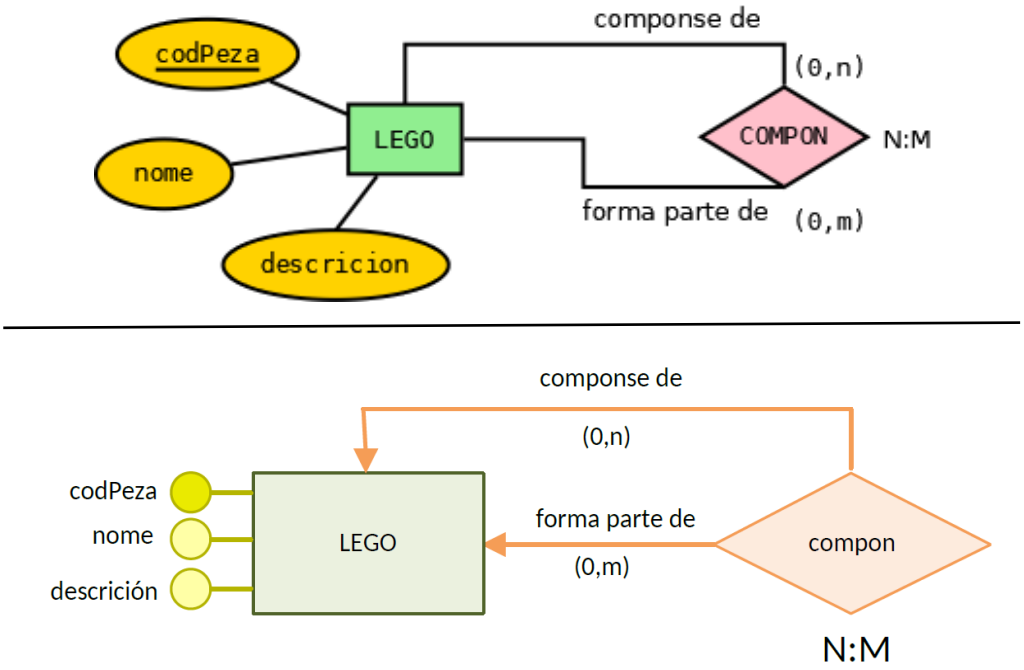
B:C, M:C

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

3.5.2.2.2 Transformación de interrelacións N:M reflexivas

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

MER



Intensión MR

LEGO (*codPeza*, nome, descrición)

B:C, M:C

B:C, M:C

COMPÓN (*codPezaCompon, codPezaFormaParte,*)

3.5.2.2.3 Transformación de interrelacións 1:N

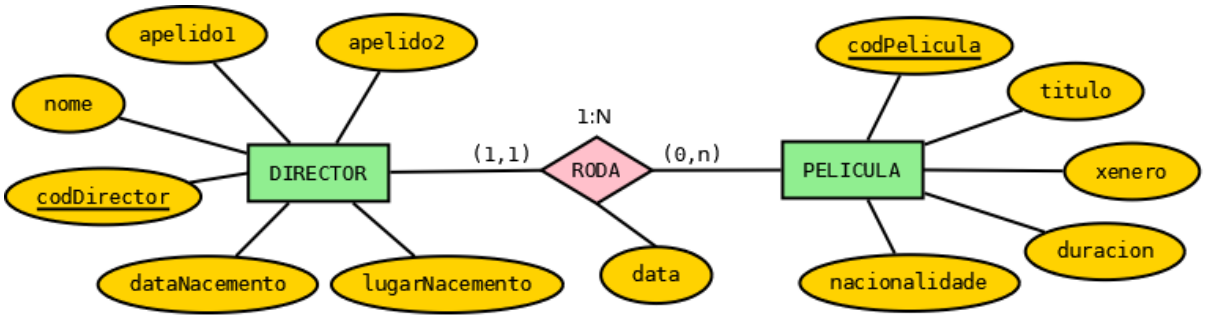
As interrelacións con tipos de correspondencia 1:N pódense transformar utilizando o modelo relacional mediante dous métodos:

I. Creación dunha restrición de integridade referencial (clave allea) que reflicta 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). Segundo o simbolismo de Chen, 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 á 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

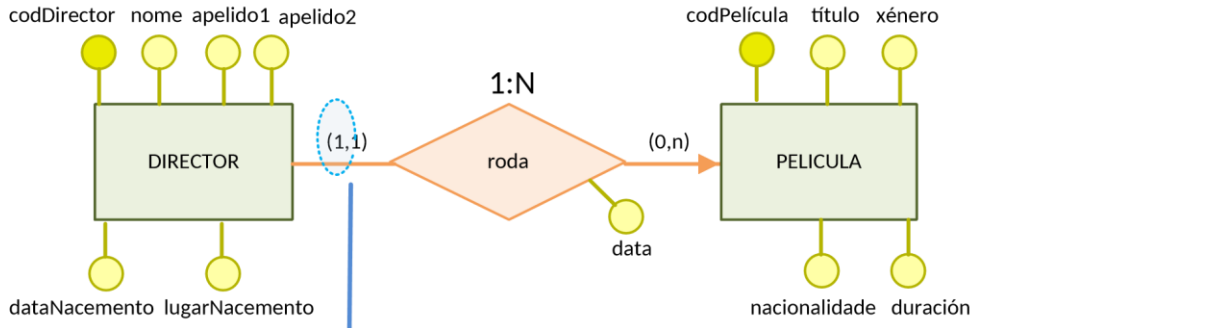
a NULO, pero se a cardinalidade mínima é 1 entón os atributos de clave allea non deben permitir o valor NULO.

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 o uso de comentarios ou mediante un nomeado especial onde se engada o nome da interrelación para recoñecer os atributos migrados.

MER



MER



Intensión MR

Obrigatorio valor NOT NULL

PELICULA (*codPelicula*, titulo, xenero, nacionalidade, duracion, data, *codDirector*)

B:C, M:C

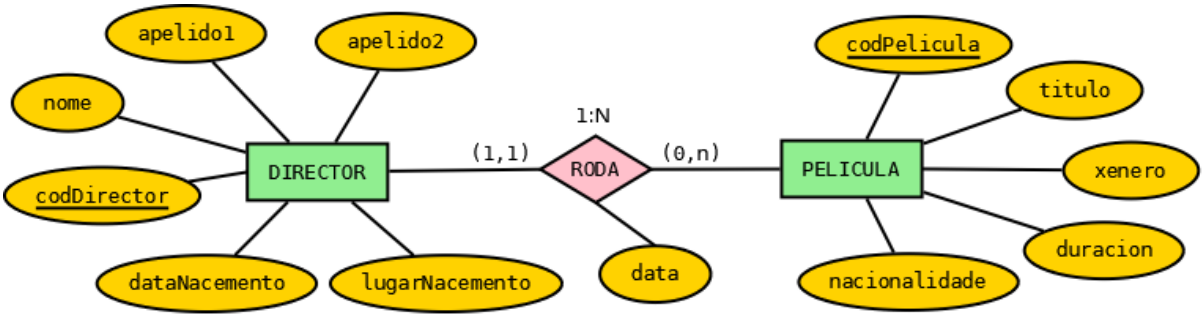
DIRECTOR (*codDirector*, nome, apelido1, apelido2, , dataNacemento, lugarNacemento)

II. Aplicar o mesmo tratamento que ás 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:

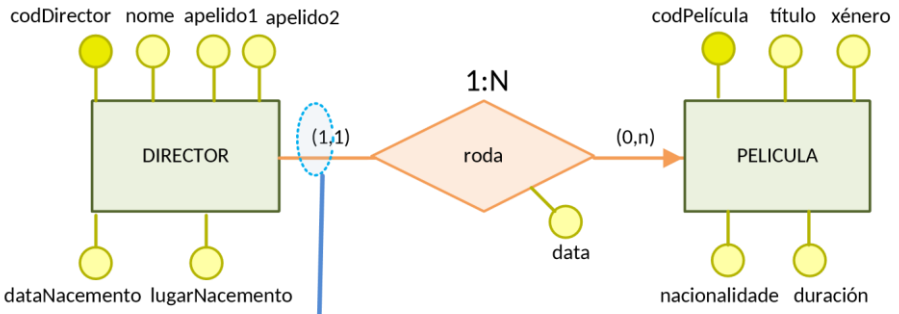
- O número de ocorrencias interrelacionadas da entidade que propaga a súa clave é moi pequeno, e polo tanto existirán moitos valores nulos na clave propagada (no exemplo, que moitas películas non teñan un director asociado).

- Prevese que a interrelación nun futuro se converterá 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 unarias ou reflexivas (no exemplo desexase coñecer certos datos da rodaxe na interrelación).

MER



MER



Intensión MR

Obrigatorio valor NOT NULL

B:C, M:C

RODA (*codPelicula*,data, *codDirector*)

B:C, M:C

PELICULA (*codPelicula*, titulo, xenero, nacionalidade, duracion)

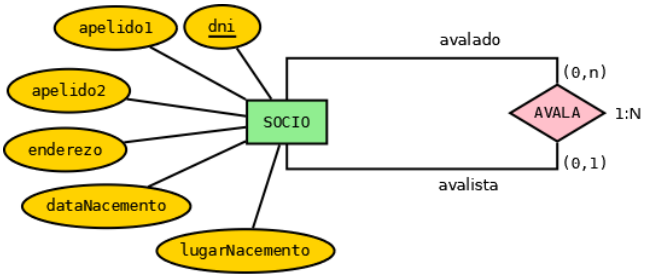
DIRECTOR (*codDirector*, nome, apelido1, apelido2, , dataNacemento, lugarNacemento)

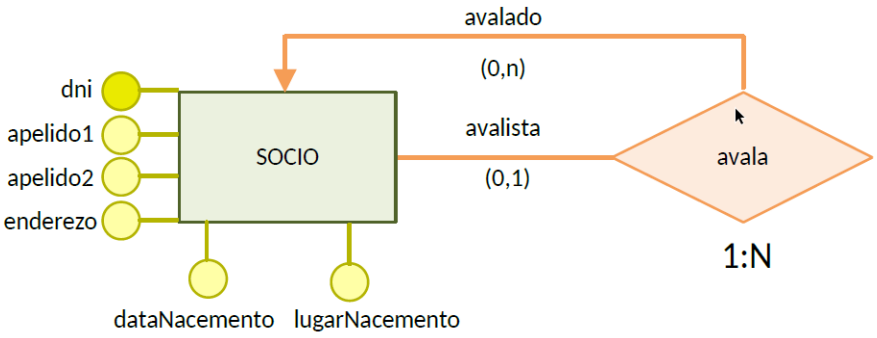
3.5.2.2.4 Transformación de interrelación 1:N reflexivas

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.

MER





Intensión MR

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

B:R, M:C

3.5.2.2.5 Transformación de interrelacións 1:1

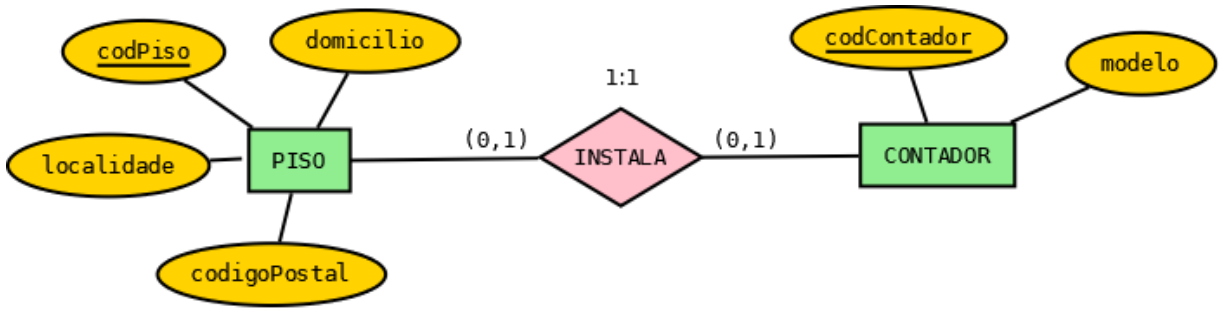
Unha interrelación de tipo de correspondencia 1:1 pódese considerar un caso particular dos tipos de correspondencia 1:N, polo que se poden aplicar as dúas opcións xa comentadas: crear unha nova táboa ou propagación de clave, tendo en conta que agora a propagación de clave pode efectuarse en ambos sentidos.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

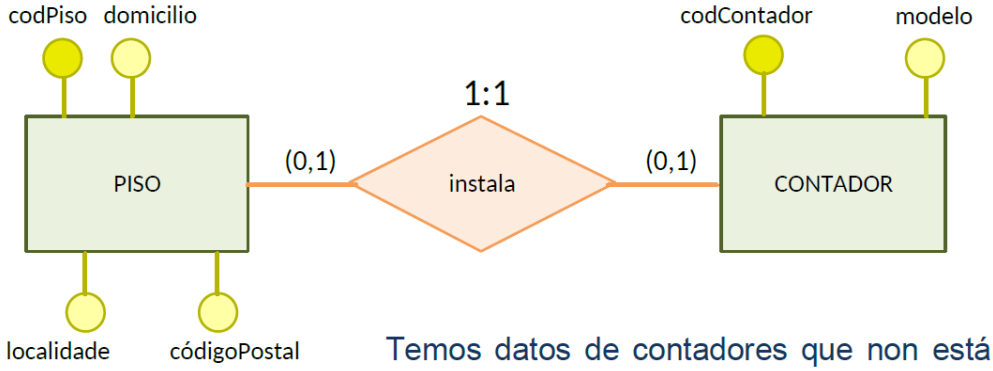
- Cardinalidades mínimas aplicadas
- Conservación da semántica do modelo
- Evitar valores nulos
- Aumento da eficiencia de procesamento

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, codigoPostal)

B:C, M:C

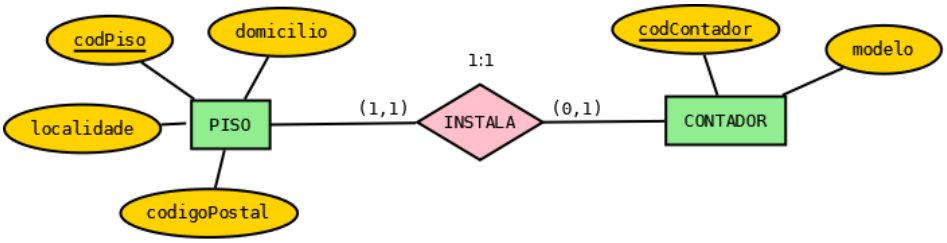
INSTALA (*codContador*, *codPiso*)

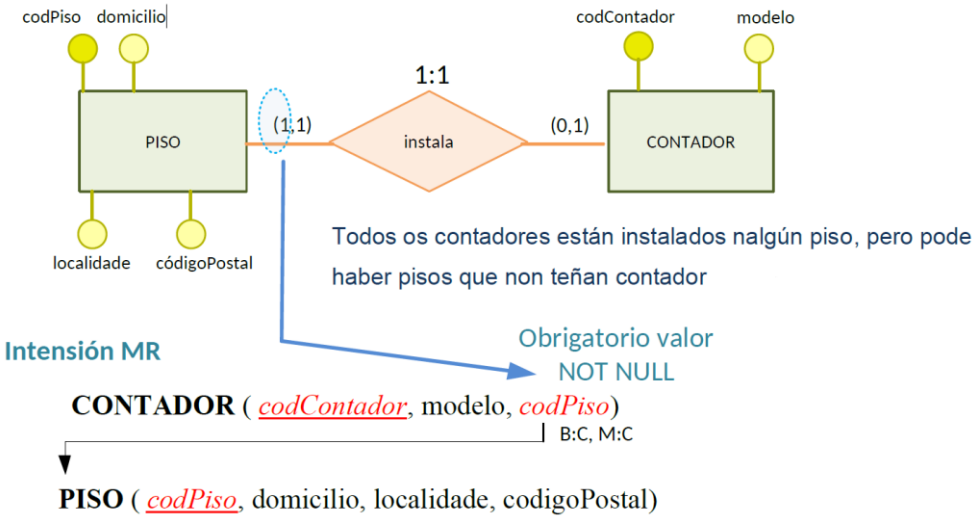
B:C, M:C

CONTADOR (*codContador*, modelo)

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 unaria con atributos, a alternativa é propagar a clave da entidade na que as súas ocorrencias participen obrigatoriamente na interrelación (propagar da relación resultante da entidade con cardinalidades (1,1) á relación resultante da entidade con cardinalidades (0,1)). Tamén se migrarían 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.

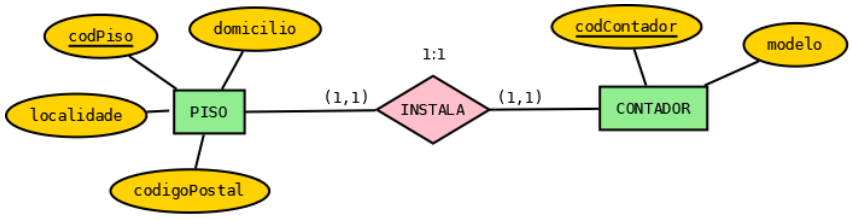
MER

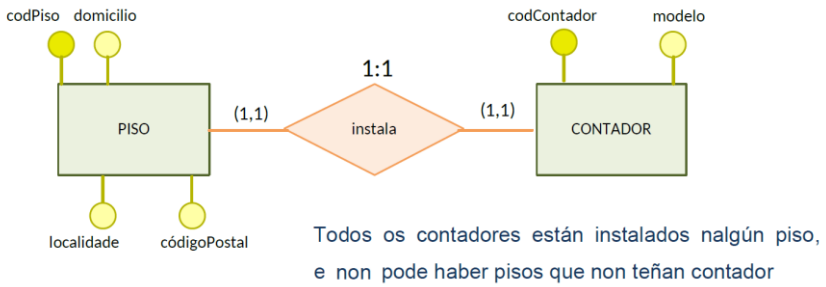




No caso de que a interrelación sexa obrigatoria para todas as entidades interrelacionadas, é dicir, ambas entidades presentan cardinalidades (1,1), poderase propagar a clave de calquera delas á relación resultante da outra; tendo en conta neste caso que é mellor que reciba a clave aquela relación que 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





Intensión MR

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

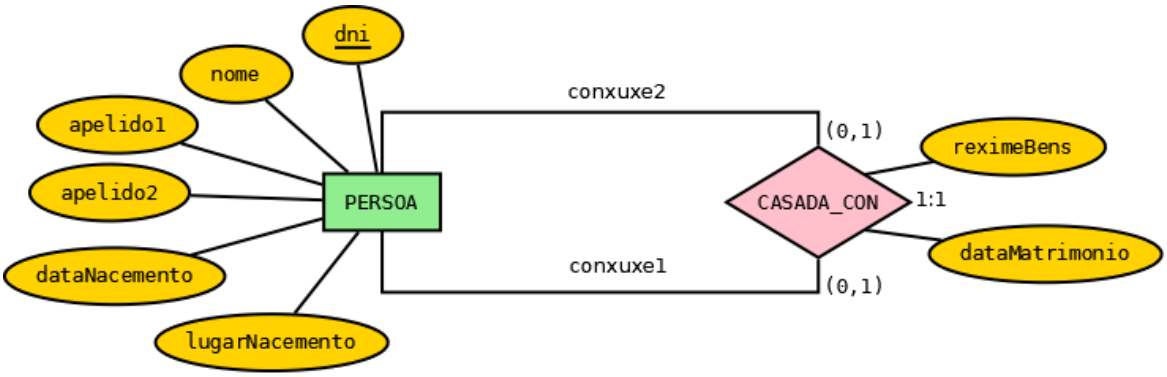
B:C, M:C

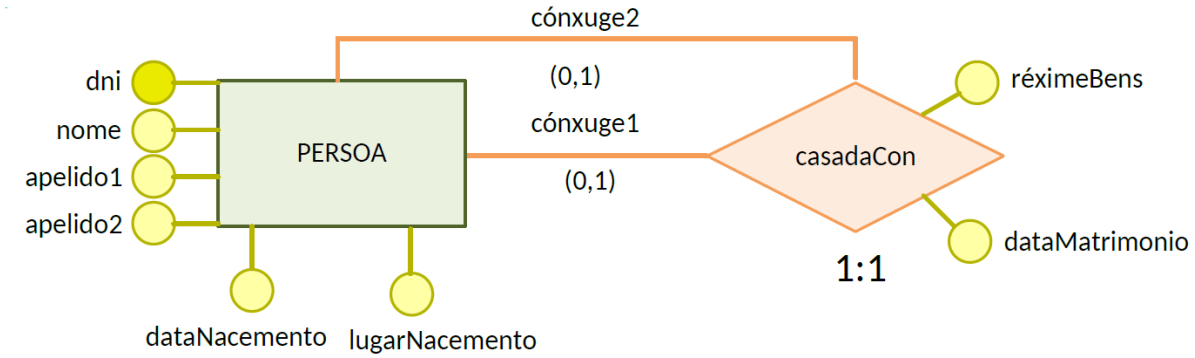
CONTADOR (*codContador*, modelo, *codPiso*)

3.5.2.2.6 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 unaria 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, dataNacemento, lugarNacemento)

B:R, M:C

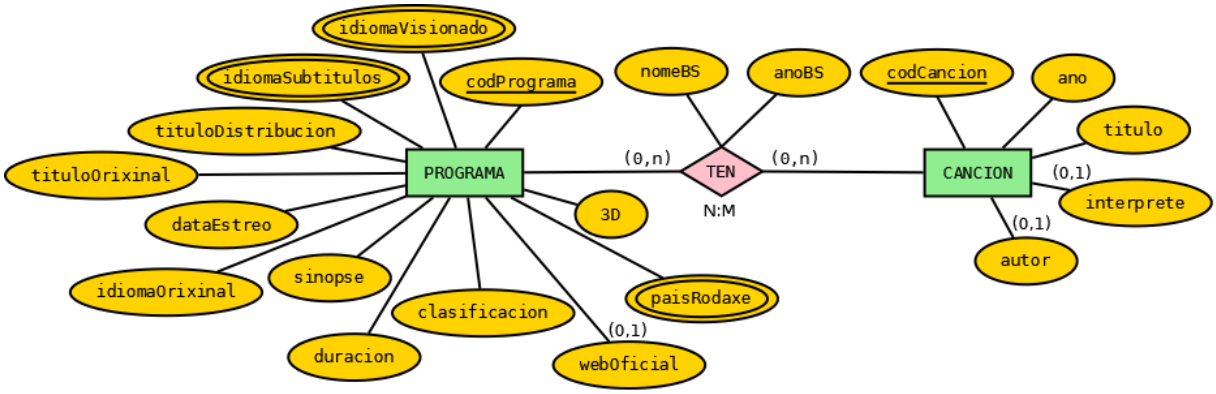
B:R, M:C

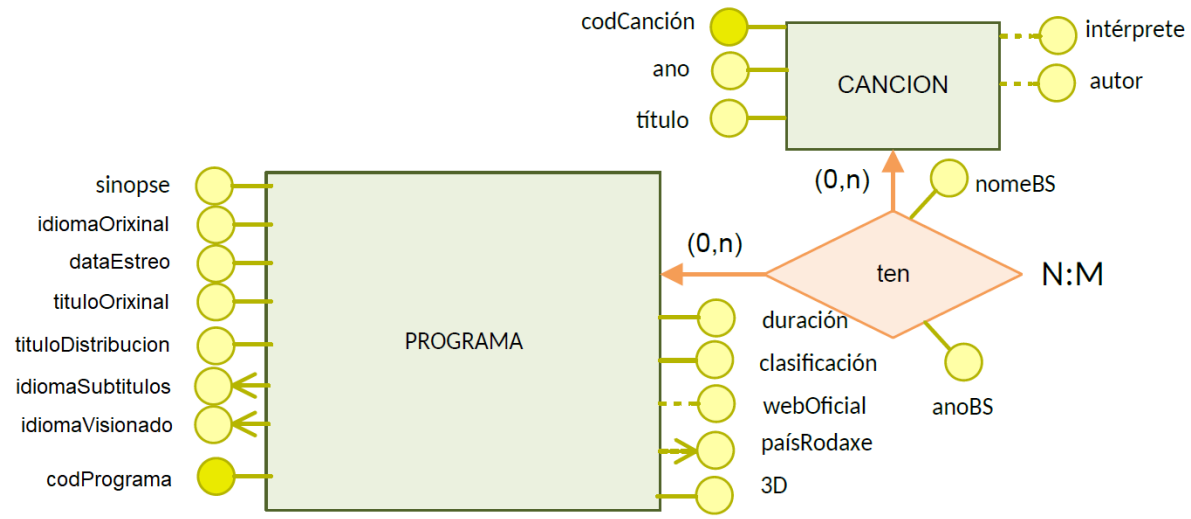
MATRIMONIO (*dniConxuge1*, *dniConxuge2*, dataMatrimonio, reximeBens)

3.5.2.3 Transformación de atributos de interrelacións

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*)

BANDA_SONORA (codBandaSonora, nomeBS, anoBS, codPrograma, codCancion)

PROGRAMA (codPrograma, tituloDistribucion, tituloOrixinal, dataEstreo, idiomaOrixinal, sinopse, duracion, clasificacion, webOficial*, 3D)

PAIS_RODAXE (codPrograma, nomePais)

IDIOMA_SUBTITULOS (codPrograma, nomeIdiomaSubtitulos)

IDIOMA_VISIONADO (codPrograma, nomeIdiomaVisionado)

3.5.3 Transformación do Modelo Conceptual MERE ao Modelo Lóxico Relacional de Codd

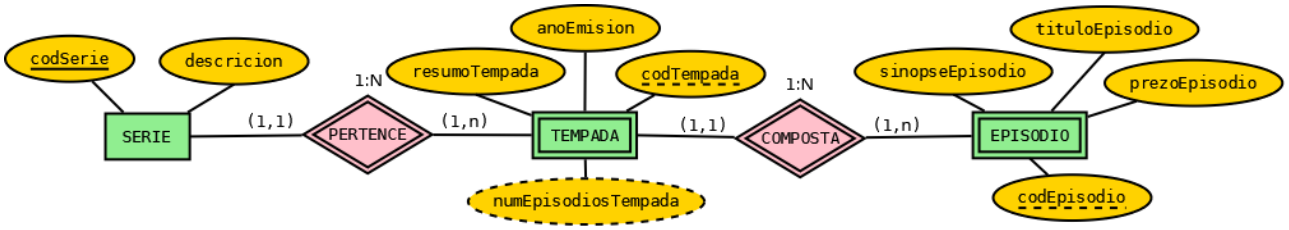
3.5.3.1 Transformación de Entidades débiles

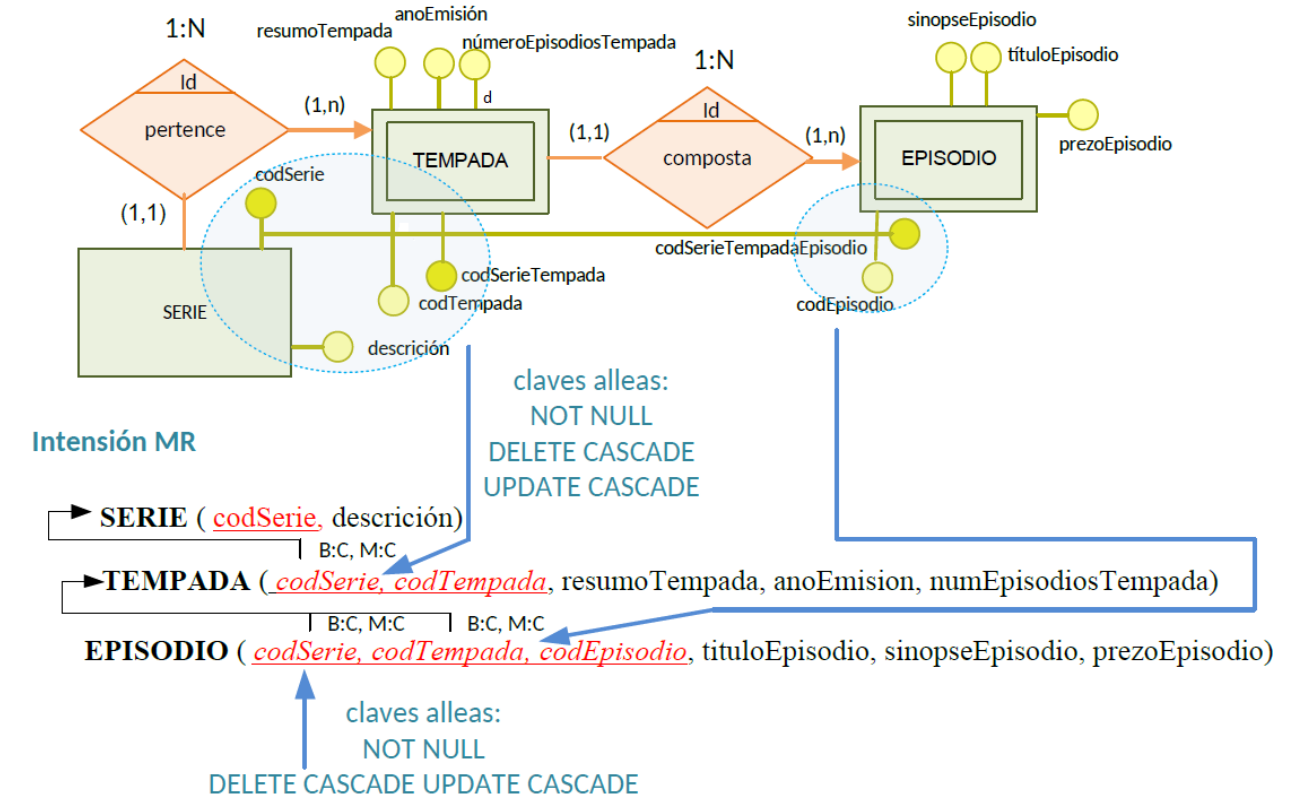
No Modelo Lóxico de datos Relacional non existe unha sentenza simple que permita indicar as dependencias en existencia ou en identificación. Poren, posto que se trata dun tipo de correspondencia 1:N que presenta determinadas restricións de cardinalidade, pódese empregar o mecanismo de propagación de clave, creando unha clave allea con NULOS non permitidos, na relación correspondente á entidade dependente.

Ademais, será preciso establecer un comportamento no caso de modificacións que obrigue a unha modificación e borrado en cascada.

No caso de que a dependencia sexa en identificación, a clave primaria da relación da entidade débil debe estar formada pola concatenación das claves propagadas das entidades participantes na interrelación máis un atributo discriminante.

MER





3.5.3.2 Transformación de xerarquías de tipos e subtipos (xeneralización ou especialización)

A definición de supertipos e subtipos emprega o mesmo concepto de herdanza que a metodoloxía orientada a obxectos, polo que, tendo mecanismos para representalos no modelo relacional, a súa semántica é máis precisa nos motores de datos obxecto relacionais.

As estratexias de transformación ao modelo relacional nun esquema conceptual formado por unha entidade supertipo e varias subtipos, directamente dependentes da primeira, están suxeitas a consideracións como as perdas semánticas, os tipos de xerarquías e cuestións de rendemento.

Destacamos as seguintes estratexias a seguir:

3.5.3.2.1 Englobar todos os atributos da entidade supertipo e os seus subtipos nunha soa relación

Esta solución é recomendable unicamente cando se dan as seguintes condicións:

- Os subtipos diferéncianse entre si en moi poucos atributos.
- As interrelacións que asocian os distintos subtipos co resto das entidades poden unificarse para todos os subtipos.
- A xerarquía é non solapada (exclusiva), é dicir, unha ocorrencia do supertipo non está nunca acompañada pola ocorrencia de máis dun subtipo.
- A xerarquía é total ou case total, é dicir, unha ocorrencia do supertipo está acompañada, na maioría dos casos, de polo menos unha ocorrencia do subtipo.

No caso de seleccionar esta transformación terase en conta o seguinte:

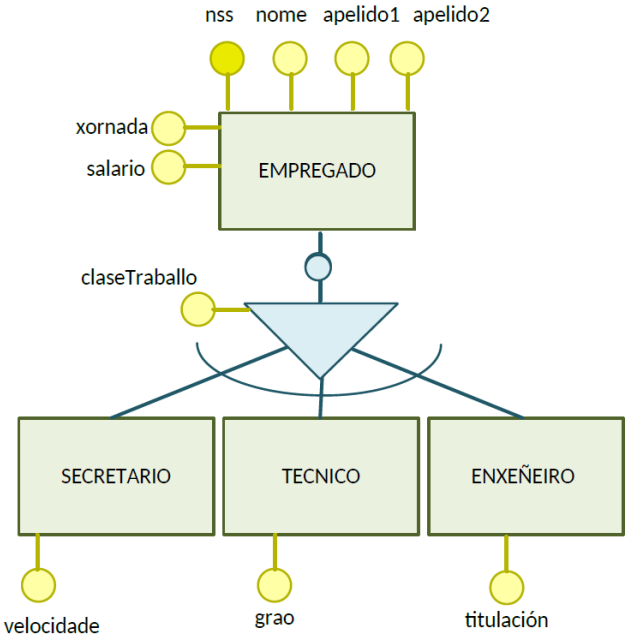
- Cando se trate dunha **xerarquía parcial**: o atributo discriminante da xerarquía (o que indica cal é o subtipo dunha ocorrencia dada) deberá admitir NULOS xa que ter este atributo a nulo indica que a ocorrencia non pertence a ningunha das subclases. Así mesmo, o valor concreto do atributo discriminante identificará a que subclase ou subclase pertence unha ocorrencia dada.
- Cando se trate dunha **xerarquía total**: o atributo discriminante da xerarquía (o que indica cal é o subtipo dunha ocorrencia dada) NON poderá admitir NULOS, xa que o seu valor identificará ao subtipo concreto ao que pertence a ocorrencia.

  Os atributos pertencentes ás subclases deben permitir a súa posta a nulos.

Se existen restricións adicionais, como condicións entre os valores dos diferentes atributos, etc. implantaranse utilizando as sentenzas e cláusula de SQL que permiten a instrumentación de restricións (verificacións ou check, disparadores ou triggers, asertos, etc.)

No que se refire á eficiencia, esta opción é a que ofrece máis velocidade no acceso aos datos dun obxecto, xa que non é preciso efectuar unións de táboas para a recuperación da información.

MER



Intensión MR

EMPREGADO (*nss*, nome, apelido1, apelido2, xornada, salario, claseTraballo, velocidade*, grao*, titulacion*)

Tamén se contempla a posibilidade de non incluír o atributo discriminante na relación, e deducir o subtipo ou subtipos aos que unha ocorrencia pertence a partir da existencia ou non dos valores dos atributos pertencentes a ditas entidades subtipo. Aínda que esta é unha solución pouco aconsellada xa que reduce o número de atributos da relación, aumenta as operacións necesarias para saber a que subtipo ou subtipos pertence unha ocorrencia.

3.5.3.2.2 Crear unha relación para o supertipo e unha que englobe todos os subtipos

Nesta estratexia, crearase unha única relación que englobará os atributos de todas as entidades subtipos que aceptarán valores nulos, e cuxa clave principal será a mesma clave principal da entidade supertipo, polo que actuaría tamén como clave allea. A inclusión do atributo discriminante non é obrigatorio, xa que a pertenza a unha das entidades subtipo coñécese pola existencia de valores nos seus atributos, pero si que é aconsellable para reducir o custo en futuros accesos.

Esta opción é recomendable unicamente cando se cumpren as seguintes condicións:

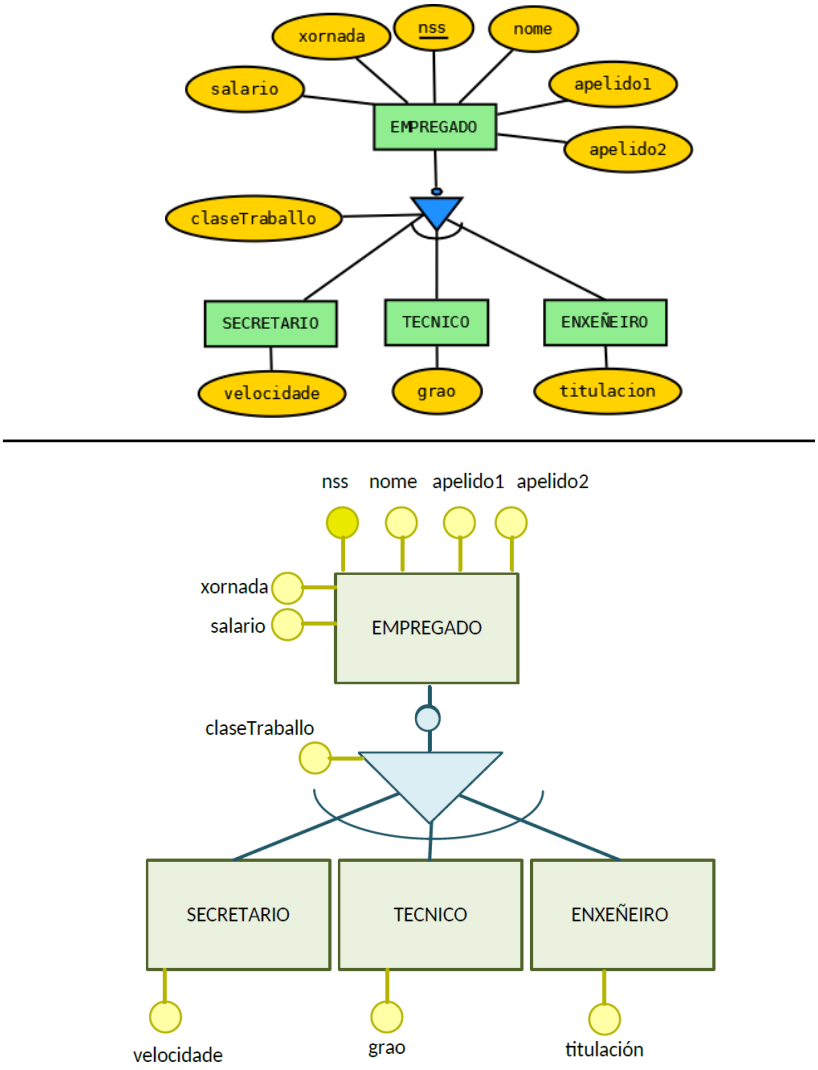
- Os subtipos diferéncianse entre si en moi poucos atributos.
- As interrelacións que asocian os distintos subtipos co resto das entidades se poden unificar para todos os subtipos.
- A xerarquía é exclusiva (non solapada), é dicir, unha ocorrencia do supertipo non está nunca acompañada pola ocorrencia de máis dun subtipo.

No caso das xerarquías parciais (non totais, isto é, existen ocorrencias do supertipo que non pertencen a ningún subtipo), será recomendable separar polo menos en dúas relacións a supertipo e o conxunto das subtipo, pois non facelo deste xeito implicaría a presenza de todos os atributos das subtipo con nulos nos casos de ocorrencia da supertipo sen ocorrencia de ningunha subtipo.

En calquera caso, a solución das dúas relacións non incorpora aumentos significativos de eficiencia respecto ás solucións que se presentan a continuación, xa que para acceder a todos os datos dun obxecto vai a requirir sempre a consulta das dúas relacións.

Dende o punto de vista do mantemento da semántica orixinal tampouco se trata dunha solución demasiado correcta, desaconsellando o seu uso.

MER



Intensión MR

EMPREGADO (*nss*, nome, apelido1, apelido2, xornada, salario)

B:C, M:C

TIPOEMPREGADO (*nss*, claseTraballo, velocidade*, grao*, salario*)

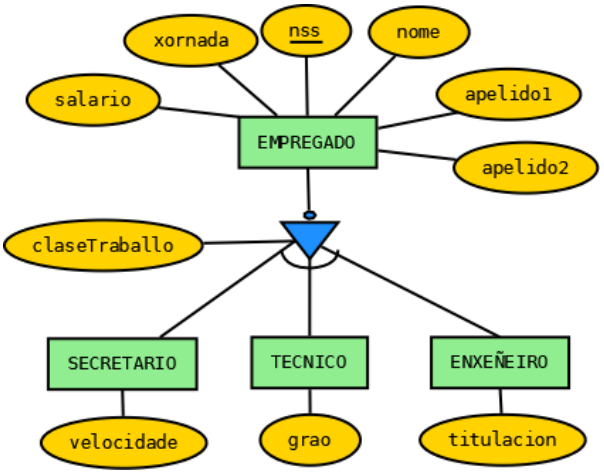
Cando non se cumpra a terceira condición existindo unha xerarquía solapada (unha ocorrencia dunha supertipo pode corresponder a unha ocorrencia de máis dun subtipo) pódese aplicar tamén esta opción, aínda que incrementa moito o custe en programación, sendo unha opción desaconsellada. Ademais, no caso de que a xerarquía sexa solapada o rendemento vai ser o menor de todos.

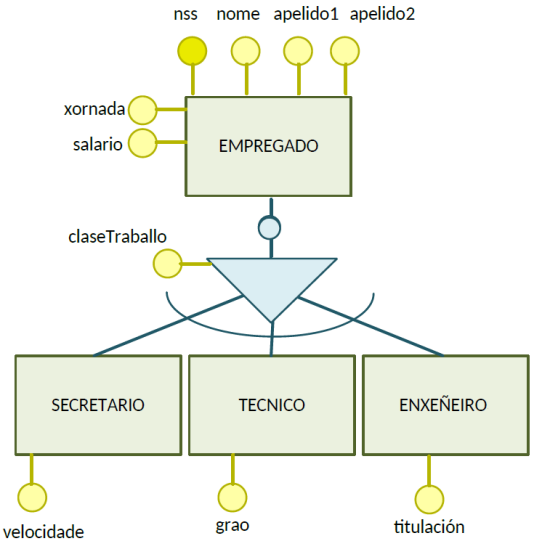
3.5.3.2.3 Crear unha relación para o supertipo e unha para cada subtipo

É a solución máis adecuada en xeral, e a máis flexible, a que permite recoller máis semántica sendo tamén a máis respectuosa co modelo conceptual orixinal. É preciso crear as restricións oportunas para que se reflicta o máis fidedignamente a semántica orixinal do modelo conceptual de datos.

No que se refire á implementación, esta opción ofrece menos velocidade que a primeira no acceso aos datos dunha relación, posto que para a recuperación da información é preciso unir varias relacións. Así mesmo, é a que máis espazo ocupa, xa que ter atributos nunha única relación sempre ocupa menos que telos repartidos entre n táboas diferentes. Sen embargo, como xa se viu, nos casos de xerarquías solapadas funciona mellor que a opción de só dúas relacións.

MER





Intensión MR

EMPREGADO (*nss*, nome, apelido1, apelido2, xornada, salario)

B:C, M:C

SECRETARIO (*nss*, velocidade)

B:C, M:C

TECNICO (*nss*, grao)

B:C, M:C

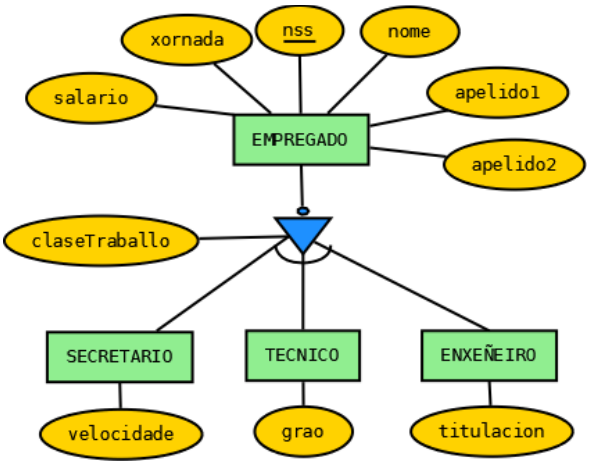
EXEÑEIRO (*nss*, tipo)

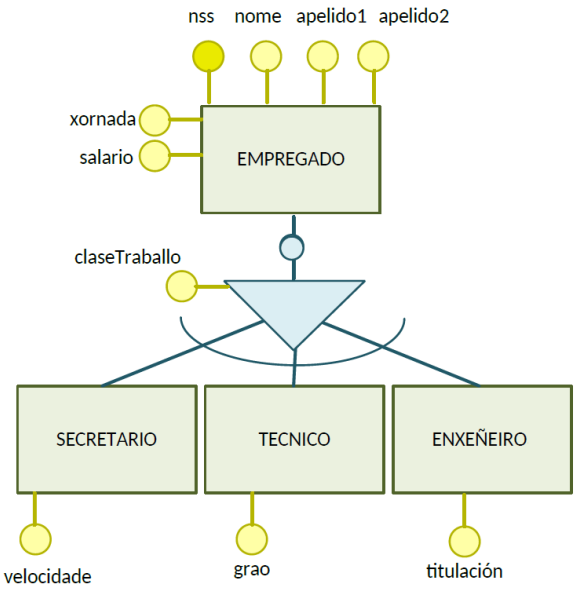
3.5.3.2.4 Crear tantas relacións como subtipos e non considerar o supertipo

Nesta opción crearase unha relación por cada subtipo existente que conterá, ademais dos atributos propios, os atributos comúns do supertipo. Optarase por esta estratexia cando se cumpren as seguintes condicións:

- Os subtipos dispoñen dun elevado número de atributos, e/ou interrelacións propias.
- Os accesos realizados aos datos dos subtipos afectan maioritariamente aos atributos comúns.
- Para unha xerarquía exclusiva disxunta, xa que de ser solapada os atributos comúns estarían a ser repetidos tendo que controlar esta redundancia para evitar inconsistencias.
- Para unha xerarquía con participación total, xa que de ser parcial xeraranse unha gran cantidade de NULOS (os atributos comúns).

MER





Intensión MR

SECRETARIO (*nss*, nome, apelido1, apelido2, xornada, salario, velocidade)

TECNICO (*nss*, nome, apelido1, apelido2, xornada, salario, grao)

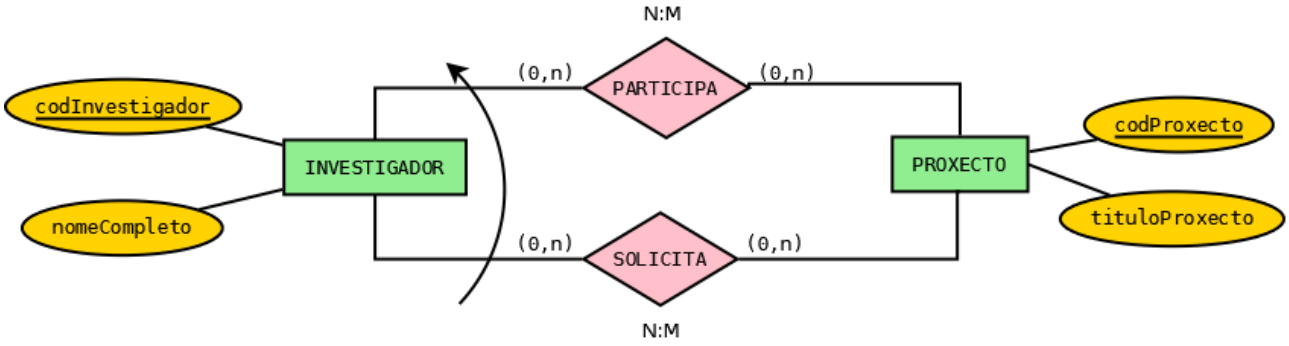
EXEÑEIRO (*nss*, nome, apelido1, apelido2, xornada, salario, tipo)

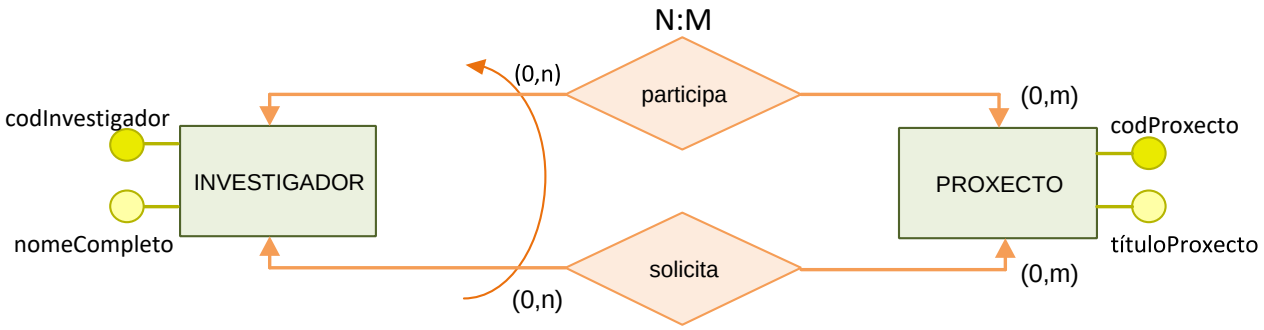
Esta solución é a que produce unha maior perda de semántica, a pesar que aumenta a eficiencia nas consultas que afectan a todos os atributos (tanto comúns como propios dun subtipo) diminuíndose noutras.

3.5.3.3 Transformación de restricións de interrelacións

A transformación de restricións de interrelacións utilizará os mesmos mecanismos que a transformación de restricións que afectan as entidades e os atributos, isto é, empregar as condicións de validación ou CHECK, asercións ou ASSERTION, disparadores ou TRIGGERS e módulos programados.

MER





Intensión MR

N:M

INVESTIGADOR (*codInvestigador*, nomeCompleto)

PROXECTO (*codProxecto*, tituloProxecto)

B:R, M:C B:R, M:C

PARTICIPA (*codInvestigador*, *codProxecto*)

B:R, M:C B:R, M:C

SOLICITA (*codInvestigador*, *codProxecto*)