



Universidade Nova de Lisboa

OMNIS CIVITAS CONTRA SE DIVISA NON STABIT

Faculdade de Ciências e Tecnologia



## Desenho de Bases de Dados; Modelo ER; Modelo Relacional

(5 Jan 2018. Aula prática 1)

### Nota introdutória

*Tendo em conta que este curso foi desenhado para 15 horas, dada a vastidão do tema, o curso terá um carácter teórico-prático e intensivo, abrangendo as diversas fases e ferramentas ligadas ao desenho e exploração das Bases de Dados, com especial incidência no rigor dos Modelos de Dados. Para um estudo apoiado sugere-se a seguinte publicação: Databases Systems, Database System Concepts – 5th Edition, Silberschatz, Korth and Sudarsham.*

### Noções Básicas

#### Entidades e Atributos

No contexto das Bases de Dados, o termo **Entidade** significa um objecto caracterizável por um conjunto de atributos. Por seu turno, o termo **Atributo** pode ser visto de forma simples como um objecto normalmente não caracterizável por outros atributos, ou por outras partes, que o descrevam. Por exemplo, a entidade *sócio* (o sócio de um vídeo-club), é caracterizável pelos seguintes atributos: *num\_sócio*, que significa o número que é atribuído ao sócio pelo vídeo-club; *BI\_sócio*, cujo significado é, obviamente, o número do BI do sócio; *nome\_sócio*, que significa o nome que figura no BI do sócio; *data\_nsc\_sócio*, que é a data de nascimento do sócio; *morada\_sócio*, *tlf\_sócio* e *sexo\_sócio* são mais três atributos cujo significado não levanta dúvidas.

Sendo o sócio uma pessoa, poderíamos à partida pensar em incluir muitos outros atributos: *sexo\_socio*, *num\_contribuinte*, *local\_nascimento*, etc.. No entanto, não é previsível qualquer interesse nesses atributos para as aplicações possíveis no contexto de um vídeo-club. A entidade *sócio* é pois caracterizada por estes atributos que, por seu turno, não se caracterizam por mais atributos, sendo completamente descritos pelo seu significado. No entanto, o significado de cada atributo deverá ser preciso, isto é, não pode ser vago. Desta maneira garante-se que o significado das entidades também será bem definido.

Um conjunto de entidades encontra-se, grande parte das vezes, numa tabela, que é uma estrutura implementável através das ferramenteas que fazem parte dum Sistema de Gestão de Bases de Dados

(SGBD). Segundo a representação E/R (Entidade/Relação), o conjunto de entidades *sócio* pode representar-se assim:

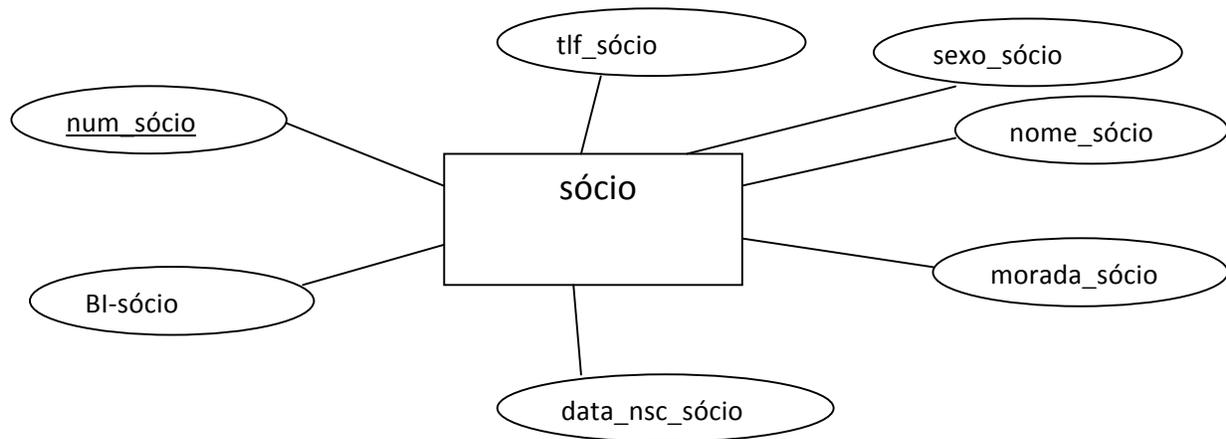


Figura 1

Na notação E/R, os **retângulos** representam conjuntos de entidades; os **losangos** representam conjuntos de relações (aborda-los-emos mais adiante); as **linhas** ligam atributos aos conjuntos de entidades e conjuntos de associações; as **elipses** representam atributos; atributos **sublinhados** representam atributos constituintes da chave primária (cencito abordado mais adiante).

### Dependências Funcionais, Superchaves e Chaves

Quando um conjunto de atributos  $\alpha$  determina univocamente um conjunto de atributos  $\beta$  considerando um conjunto de entidades, diz-se que a **dependência funcional**  $\alpha \rightarrow \beta$  é verdadeira nesse conjunto de entidades. Por exemplo, no conjunto de entidades *sócio*, o atributo *num\_sócio* determina univocamente *nome\_sócio*, isto é, não existem 2 nomes de sócio diferentes para o mesmo número de sócio, qualquer que seja o conjunto de entidades. Isto quer dizer que a dependência funcional  $num\_sócio \rightarrow nome\_sócio$  é verdadeira (ou existe) no conjunto de entidades *sócio*. Analogamente,  $num\_sócio \rightarrow tlf\_sócio$  também é verdadeira em *sócio*, como se compreende. Assim, facilmente se conclui que  $num\_sócio \rightarrow tlf\_sócio$ ,  $nome\_sócio$  também é verdadeira em *sócio*. Isto é, se o atributo *num\_sócio* determina separadamente vários atributos, também os determina simultaneamente. Também facilmente se percebe que  $nome\_sócio \rightarrow tlf\_sócio$  é verdadeira, já que a cada *nome\_sócio* só corresponde um *tlf\_sócio* (por hipótese). Também se compreende que qualquer atributo se determina a si próprio, isto é,  $\alpha \rightarrow \alpha$ , dependência funcional também conhecida por **trivial**.

No entanto, quer  $tlf\_sócio \rightarrow nome\_sócio$  quer  $morada\_sócio \rightarrow BI\_sócio$ , entre muitas outras dependências funcionais, não são verdadeiras em *sócio* já que, considerando o primeiro caso, podem existir duas entidades com diferentes valores para o atributo *nome\_sócio* mas com o mesmo valor para o atributo *tlf\_sócio*: dois sócios que vivam na mesma casa podem ter o mesmo número de telefone.

No caso particular em que um conjunto de atributos  $\alpha$  determina todo o conjunto de atributos que constitui a entidade (conjunto normalmente designado por **esquema** R), isto é,  $\alpha \rightarrow R$ , diz-se que  $\alpha$  é uma **superchave** no conjunto de entidades. Por outras palavras, quando um conjunto de atributos determina univocamente cada entidade dentro do conjunto de entidades, diz-se que esse conjunto de atributos é uma superchave do conjunto de entidades. Muitas vezes esse conjunto de atributos é constituído apenas por um atributo, como é o caso de *num\_sócio* como facilmente se percebe. É claro que se juntarmos o atributo *data\_nsc\_sócio* ao atributo *num\_sócio*, o conjunto formado por estes dois atributos também determina univocamente cada entidade, sendo por isso outra superchave. Há no entanto, um caso particular de superchaves: as superchaves minimais, ou seja, aquelas que deixarão de ser superchaves, casos lhe seja retirado qualquer dos atributos. Às superchaves minimais chamam-se **chaves**.

Assim, facilmente se percebe que *num\_sócio* é uma chave, uma vez que  $num\_sócio \rightarrow num\_sócio, nome\_sócio, BI\_sócio, tlf\_sócio, morada\_sócio, data\_nsc\_sócio$  se verifica; ou ainda e de forma abreviada,  $num\_sócio \rightarrow R$  (como já se disse, o símbolo R reserva-se para indicar todo o conjunto de atributos). Pelas mesmas razões, também *BI\_sócio* é uma chave. Aos atributos *num\_sócio* e *BI\_sócio* chamam-se chaves candidatas. Entre os dois escolher-se-á aquele que fisicamente for mais curto e chamar-se-á **chave primária**, por vezes também conhecida por **chave designada**.

Convida-se o leitor a reflectir sobre que outros conjuntos de atributos no esquema *sócio* poderiam também ser chaves. Será que o conjunto *nome\_sócio, morada\_sócio* pode ser uma chave?

### **Desenho de um Modelo de Dados utilizando Diagrama E/R. Problema 1**

Suponhamos que pretendemos criar uma modelo de dados e posteriormente a correspondente BD sobre um vídeo-clubes.

Além do sócio, de que já falámos, existem filmes para alugar aos sócios. Os filmes caracterizam-se pelo nome que os identifica, o ano de lançamento, o preço por dia de aluguer, o número de dias de aluguer «sem multa» e o preço por dia para além da data previsível de entrega. No filme participa pelo menos um actor, bem como pelo menos um realizador. Admitamos que um filme pertence sempre a um só género (cómico, dramático, etc.) e pertence sempre a uma só editora que é identificada pelo seu nome. Os filmes podem ter uma ou mais cópias. As cópias pertencentes a um filme têm números de cópia seguidos a começar em 1. Por outras palavras, num filme do qual haja 3 cópias, estas terão como números de cópia, 1, 2 e 3. Alugam-se cópias dos filmes aos sócios. Um aluguer é feito sempre numa data e também é caracterizado por uma data previsível de devolução da cópia. No acto de devolução é registado o estado de conservação da cópia; neste acto poderá haver lugar a multa por entrega tardia.

Desenhe um Modelo de Dados para este vídeo-clubes. Tenha em conta que **os dados devem organizar-se de forma independente das aplicações com os quais elas venham a trabalhar**.

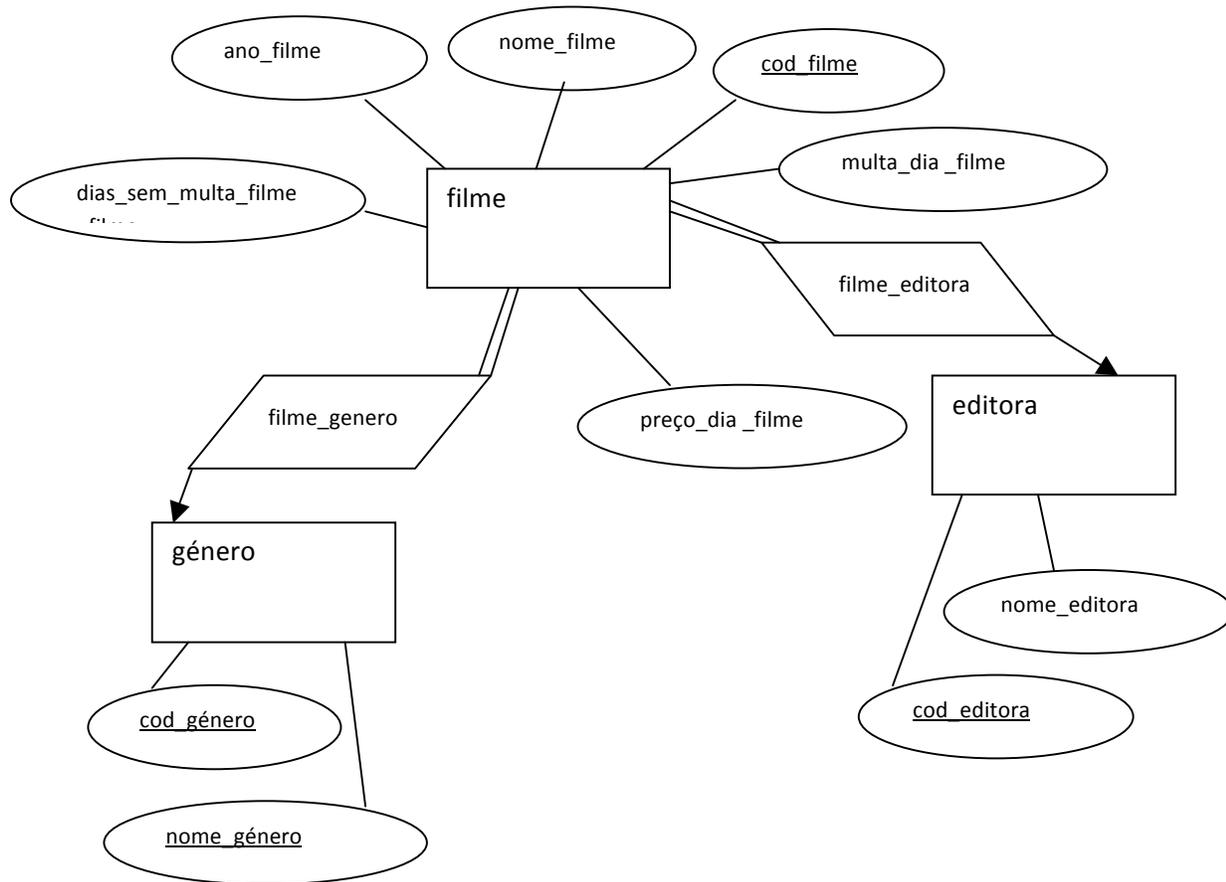


Figura 2

O atributo *nome\_filme* identifica univocamente o filme, ou seja, é uma chave candidata no conjunto de entidades/esquema, já que podemos admitir que não existem dois filmes com o mesmo nome. Porém, *nome\_filme* é um atributo que pode ocupar 30 ou 40 caracteres, não sendo por isso aconselhável para ser chave primária. Nestes casos é costume criar-se um código (*cod\_filme*), numérico e bem mais curto que, identificando também univocamente cada entidade, é escolhido para chave primária. A mesma técnica é aplicada nos conjuntos de entidades *editora* e *gênero*, isto é, foram criados os atributos *cod\_editora* e *cod\_gênero*.

Há uma **relação/associação** de **muitos-para-um** no sentido de *filme* para *gênero*; o losango representa a relação; a seta indica o sentido da associação muitos-para-um. Por outro lado, a participação de *filme* em *filme\_gênero* é total, isto é, cada entidade particular em *filme* (cada filme concreto) pertence sempre a um gênero concreto; o traço duplo representa essa participação de totalidade.

Pelo diagrama, vê-se que, analogamente, há outra relação/associação de muitos-para-um no sentido de *filme* para *editora*.

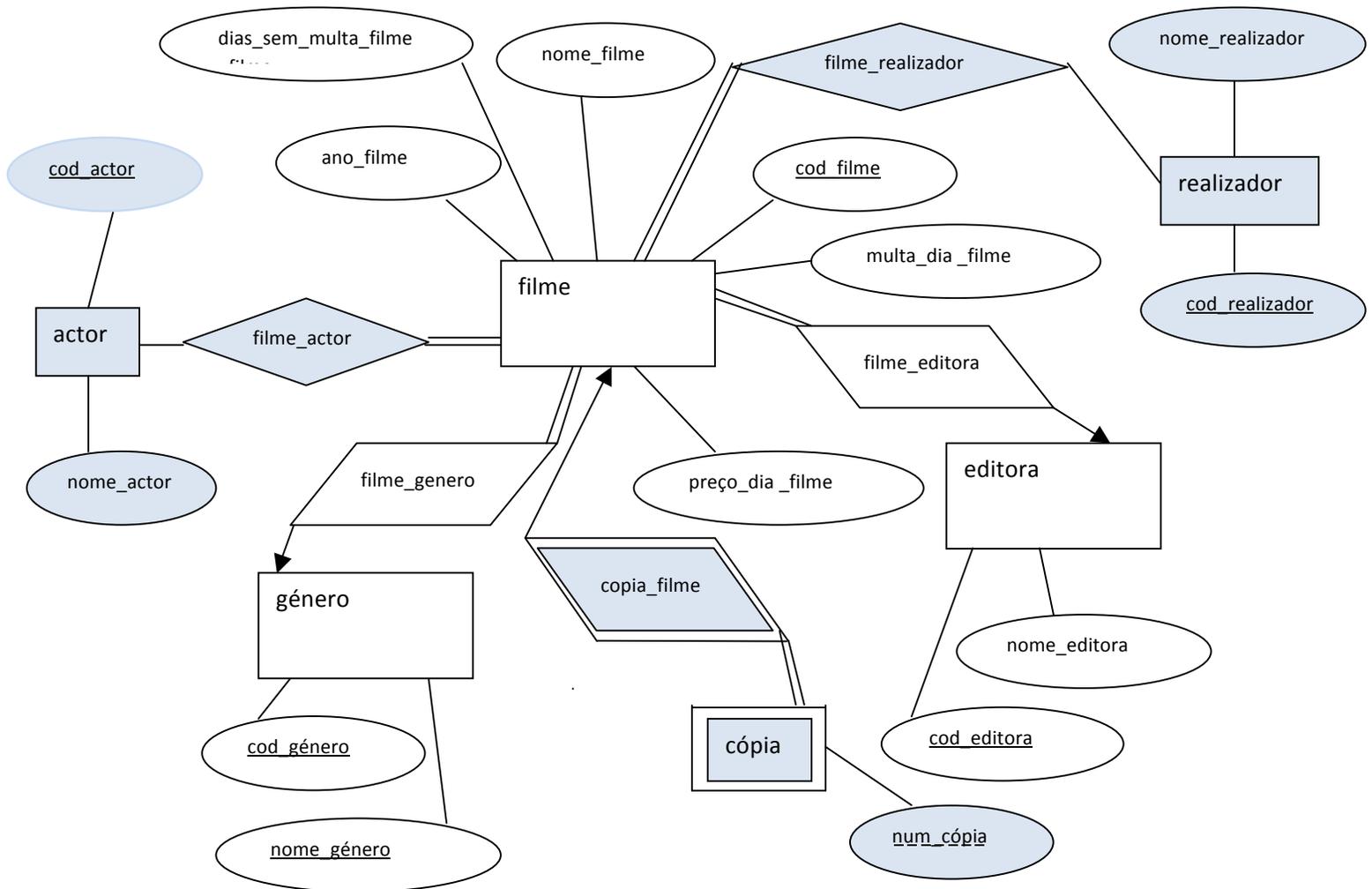


Figura 3

No diagrama, as partes a sombreado não fazem parte da notação DER, servem apenas para pôr em relevo o que acabou de ser acrescentado ao DER anterior.

Assim, há uma relação/associação entre *filme* e *actor*. A ausência de seta em ambos os sentidos indica que a associação é de **muitos-para-muitos**. No entanto, a participação de *filme* em *filme\_actor* é total, isto é, cada filme tem que ter sempre pelo menos um actor. A relação *filme\_actor* terá como chave candidata (e primária) a combinação do atributo que é chave primária em *filme* e o atributo que é chave primária em *actor*, isto é, (cod\_filme, cod\_ator). A

ordem pela qual são dispostos os atributos constituintes da chave pode ser à partida qualquer uma, no entanto como veremos mais adiante, podemos optar pela disposição que favoreça as pesquisas mais procuradas.

Análoga à relação/associação *filme\_actor*, existe outra entre *filme* e *realizador*; as considerações a ter em conta neste caso, são idênticas. Ver diagrama da figura 3.

Segundo as chamadas *especificações funcionais*, isto é, o enunciado do problema, os números de cópia serão atribuídos a partir de 1 para cada filme; isto quer dizer que no conjunto de entidades *cópia* poderão existir várias entidades com o mesmo valor para o atributo *num\_cópia*, o que impossibilita este atributo de, sozinho, ser chave primária. Nestes casos, quando no conjunto de entidades, o conjunto de atributos que surgem, digamos, «naturalmente», não é suficiente para formar chave, o conjunto de entidades é considerado **fraco** e o duplo traço que delimita o rectângulo *cópia* caracteriza isso. Nestes casos, para obter uma chave primária, é necessário juntar ao chamado atributo **discriminante** --- neste caso o *num\_cópia* --- outros atributos que façam parte de outros conjuntos de entidades ligados a este conjunto. Assim, como sabemos, toda a cópia pertence a um e um só filme --- ver duplo traço de *cópia* para *cópia\_filme* e seta de *cópia\_filme* para *filme* --- então a chave candidata (e primária) inclui o atributo que é chave em *filme*, ou seja, *cod\_filme*, para além do atributo discriminante, *num\_cópia*, como já foi dito; esta inclusão está representada pelo «duplo losango» de *cópia\_filme* na mesma figura 3.

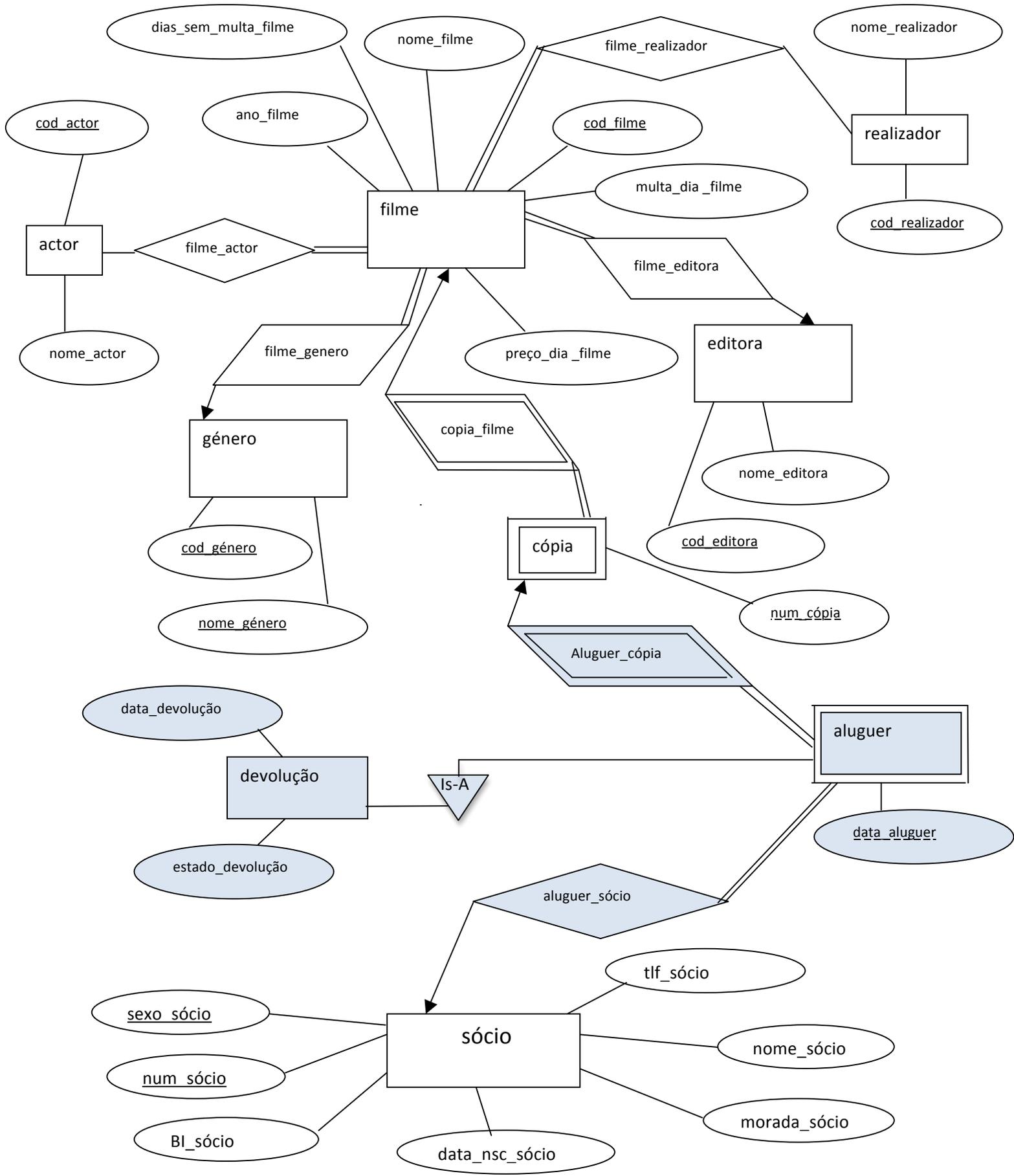


Figura 4

Segundo o enunciado, relativamente ao conjunto de entidades aluguer surge naturalmente a *data\_aluguer*, atributo que, sozinho não é suficiente para constituir a chave, já que existirão vários alugueres feitos na mesma data. Assim, também este conjunto pode ser considerado como fraco --- atenção que o adjectivo fraco apenas se deve à notação DER; não quer dizer que se esteja perante um conjunto de entidades de menor importância como por sugestão podemos ser levados a pensar. Assim, facilmente se compreende que necessitamos dos atributos que, combinados, identifiquem univocamente qualquer entidade em *aluguer* (qualquer aluguer). Em consequência disso, a relação *aluguer\_cópia* está representada com um duplo losango e a chave primária de aluguer será constituída portanto pela combinação de 3 atributos: (*cod\_filme,num\_cópia,data\_aluguer*); também aqui, a ordem dos atributos não é importante nesta fase.

Naturalmente que, todo o aluguer é feito a um sócio, daí a relação de muitos-para-um de *aluguer* para *sócio*. Ver figura 4.

*Convida-se o leitor a responder à seguinte pergunta: porque não substituir na chave do aluguer, os atributos que são chave de cópia pelos atributo que é chave de sócio?*

Continuando com o nosso modelo de dados, agora atendendo ao evento da devolução do aluguer, aquilo que é de uma forma simples, a entrega da cópia física numa dada data. Poder-se-ia pensar em acrescentar um atributo, *data\_devolução*, ao conjunto de atributos de *aluguer*. Porém, embora esta seja uma solução intuitiva, não é a solução mais robusta. Na verdade, se optássemos por ela, o atributo *data\_devolução* ficaria por preencher na maior parte do tempo, o que é de evitar. Além do mais, o evento devolução tem outros atributos, digamos naturais, entre eles o estado de conservação da cópia na altura em que esta foi devolvida; isso implicaria ter mais um atributo *estado\_devolução* em *aluguer*, que também ficaria a vazio na maior parte do tempo; tal como o atributo *data\_devolução*, também o *estado\_devolução* seria preenchido apenas quando a cópia fosse devolvida ---voltaremos a este assunto mais à frente---. Posto isto, a solução mais robusta, e por isso aconselhável, é considerar que *devolução* é um conjunto de entidades.

Acontece que, entre *aluguer* e *devolução* a cardinalidade é de **um-para-um**, com a particularidade de a uma *devolução* corresponder sempre um *aluguer*. No entanto, nem a todo o aluguer corresponde uma devolução já que a devolução só ocorre passados dias depois do aluguer. Nestes casos de um-para-um o conjunto de entidades com a *participação total* herda como sua chave primária, a chave primária do outro conjunto de entidades. Esta circunstância configura uma estrutura conhecida por *1s-A* (é um); mais concretamente, uma devolução é um caso particular de aluguer, ou seja, um tipo de aluguer. Ver figura 4.

### Passagem do modelo de dados em DER para esquema de tabelas

A passagem para o esquema de tabelas a partir do DER faz-se com regras simples. Grosso modo, a cada conjunto de entidades corresponde uma tabela. Começamos então por definir as tabelas que não estão em dependência de muitos-para-um a partir delas; por exemplo as tabelas sócio, género, editora, actor e realizador.

**Sócio=(num\_sócio, nome\_sócio, bi\_sócio, data\_nsc\_sócio, morada\_sócio, tlf\_sócio, sexo\_sócio)**

**Género=(cod\_género, nome\_genero)**

**Editora=(cod\_editora, nome\_editora)**

**Actor=(cod\_actor, nome\_actor)**

**Realizador=(cod\_realizador, nome\_realizador)**

Nestes casos, basta incluir como campos/atributos da tabela, os mesmos atributos que no DER fazem parte do conjunto de entidades, ignorando as ligações a outras relações. Os atributos a sublinhado são as chaves primárias. Entre as chaves candidatas *num\_sócio* e *bi\_sócio*, foi escolhida a primeira como chave designada (primária) por ser mais curta.

Vejamos agora o caso da tabela filme.

**Filme=(cod\_filme, nome\_filme, ano\_filme, preço\_dia\_filme, dias\_sem\_multa\_filme, multa\_dia\_filme, cod\_género, cod\_editora)**

Neste caso, para além dos atributos que, representados em elipses no DER rodeiam o rectângulo *filme* (tal como no caso anterior para *sócio*, *género*, *editora*, *actor* e *realizador*) há que incluir como atributos, aqueles (atributos) que funcionam como chaves primárias nos conjuntos de entidades com os quais existe uma relação de muitos-para-um a partir de *filme*. Daí que a tabela filme inclua também os atributos *cod\_génro* e *cod\_editora*. Na verdade, cada filme concreto tem exactamente uma editora concreta e por isso, o atributo que é a chave primária da editora (o «apontador» para a editora) pode e deve figurar nos atributos da tabela filme. Um raciocínio análogo aplica-se relativamente à chave primária da tabela género que também é «importado» como atributo para a tabela filme.

Note-se que, à relação *filme\_género* não corresponde nenhuma tabela. Em rigor, ela poderia existir com a estrutura *filme\_género*({cod\_filme,cod\_género}), isto é, teria a mesma chave que

tem a tabela filme; e por isso, no chamado **modelo simplificado de tabelas**, os atributos *não chave* desta tabela são incluídos na tabela filme. Aplica-se o mesmo raciocínio relativamente à relação *filme\_editora* e tabela filme\_editora.

No entanto, existe uma relação de cardinalidade diferente (muitos\_para\_muitos) de *actor* para *filme*. Este tipo de relação origina sempre uma tabela adicional que recebe normalmente o mesmo nome da relação; neste caso filme\_actor. Os atributos desta tabela são os atributos que constituem as chaves de cada conjunto de entidades que está na origem desta relação de muitos-para-muitos e, a reunião/combinção de todos estes atributos contituem a chave desta nova tabela. Ou seja:

**Filme\_actor=(cod filme,cod actor)**

Relativamente à relação *filme\_realizador* aplica-se o mesmo raciocínio; daí a tabela:

**Filme\_realizador=(cod filme,cod realizador)**

De notar que, poderíamos admitir um atributo que retratasse em texto livre o papel do actor concreto no filme concreto (papel\_filme\_actor). Esse atributo seria acrescentado à tabela filme\_actor mas não faria parte da chave.

Relativamente à tabela cópia, existem apenas os atributos *num\_cópia*, que é o chamado atributo discriminante como já foi referido, e *cod\_filme* (a chave primária na tabela filme; ver a seta e o seu sentido na relação *cópia\_filme*). Este último atributo irá completar a chave primária nesta tabela. Na verdade, uma cópia concreta pertence sempre a um dado filme:

**Cópia=(cod filme,num cópia)**

Relativamente à tabela aluguer, tendo em conta que deriva dum conjunto de entidades fracas, esta terá como chave o atributo discriminante *data\_aluguer* e os atributos que constituem a chave primária da tabela cópia. Para além destes, existirá também o atributo *num\_sócio*, não integrante aqui na chave de aluguer, mas que é chave na tabela sócio; ver a seta e o seu sentido na relação *aluguer\_sócio* na figura 4.

**Aluguer=(cod filme,num cópia,data aluguer,num sócio)**

Por fim, a tabela devolução, que decorre do conjunto de entidades com o mesmo nome, terá como chave a mesma chave da tabela aluguer, já que existe uma relação de um\_para\_um entre os dois conjuntos de entidades mas com participação total por parte da devolução. Esta tabela terá ainda como atributos, *data\_devolução* e *estado\_devolução*; estes já não integrantes da chave da tabela.

**Devolução=(cod filme, num cópia, data aluguer, data devolução, estado devolução)**

### **Exercício 1**

Crie um Modelo de Dados para consultório médico. Considere consultas, doentes, médicos e especialidades (estomatologia, pneumologia, etc.). Cada médico pode ser especialista em mais do que uma especialidade. Nas consultas os médicos receitam medicamentos (um ou mais). Existem também marcações de consultas.

### **Exercício 2 (parte do projecto para avaliação)**

Juntamente com o seu grupo de trabalho, crie um modelo de dados sobre um assunto à vossa escolha. O tema escolhido deverá ser aprovado pelo docente deste curso.