

Base de Dados

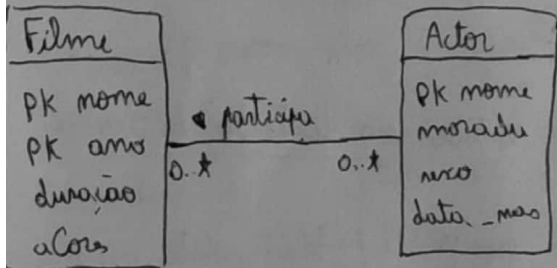
CREATE TABLE < nome > (
 < lista de elementos >
);

Tipos de Elementos

- INT ou INTEGER
- REAL ou FLOAT
- CHAR(m) , VARCHAR(m)
- BOOLEAN
- Date ('yyyy-mm-dd')
- TIME ('hh:mm:ss')

Associação entre classes

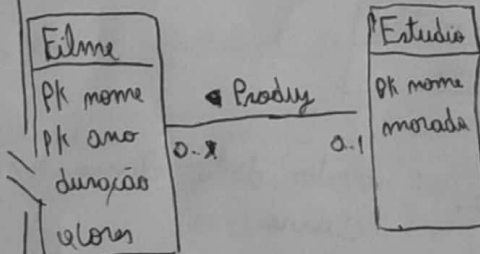
muitos - muitos



Faz-se a união das PK

Participa (nome Filme , ano Filme , nome ator)

muitos - um



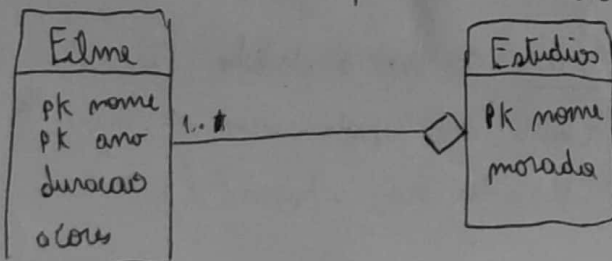
Escolhe-se o "muitos" como chave

Produz (nome Filme , ano Filme , nome Est.)

um - um

Escolhe-se uma das duas

- Agregação : associação com o significado que objetos de um dos lados podem "ser possuídos" ou "fazer parte de" objetos do outro lado.

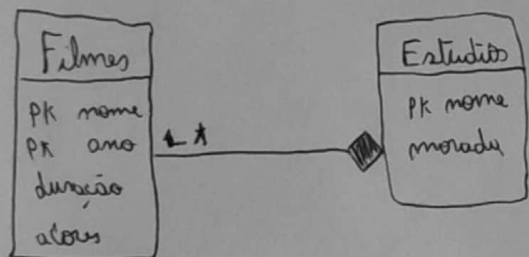


0..1

⇒ Um filme pode ser produzido por um estúdio. (O estúdio é "dono do filme")

1.

- Composições: Semelhante à agregação mas a posse é obrigatória



⇒ Um filme tem de ser forçosamente produzido por um estúdio
 ⇒ mete-se no lado da classe que é "dona de";
 significa 1..1

- Subclasses: Uma classe pode ter subclasses

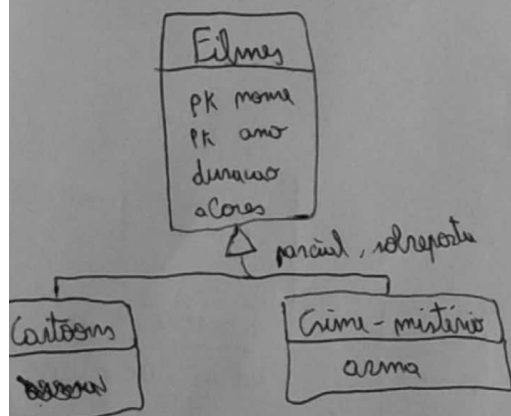
ex: Um cartoon é um tipo de Filme, Crime-mistério igual

• Nota: nunca se usa PK

Subclasses → completas ou parciais: completa se cada objeto da superclasse está numa das subclasses, parcial caso contrário

→ disjuntas ou sobrepostas: disjunta se um objeto não pode estar em mais do que uma subclasse, sobreposta caso contrário

- Parcial, sobreposta



- Sobreposta porque um filme pode ser simultaneamente um cartoon e crime-mistério

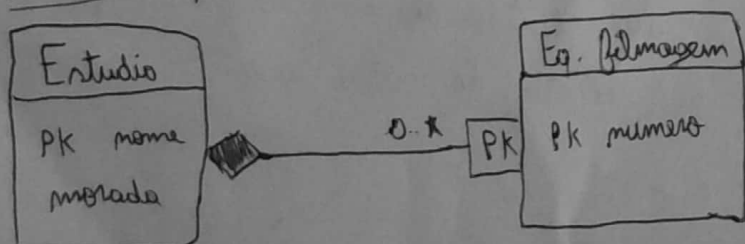
Filmes (nome, ano, duração, cores, arma)

Classes de Suporte: objetos desta classe não fazem sentido no contexto de outra classe

ex: Um estúdio pode ter várias equipas de filmagem: equipa 1, 2, 3...

Uma equipa de filmagem é uma classe de suporte (fraca) a um estúdio

Uma equipa de filmagem só pode ser identificada se for "buscar" a chave do estúdio respetivo

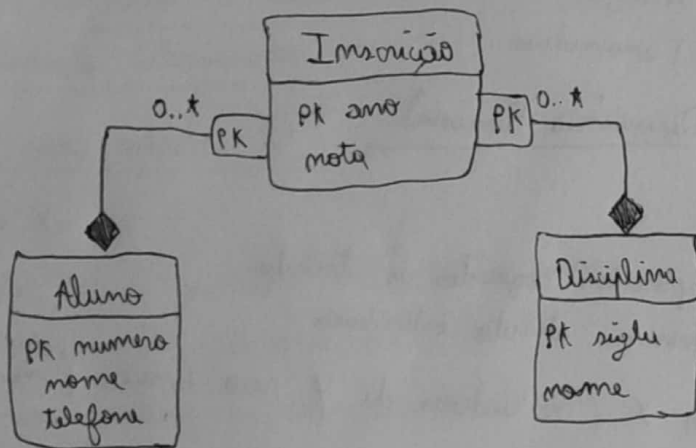


Estúdio (nome, morada)

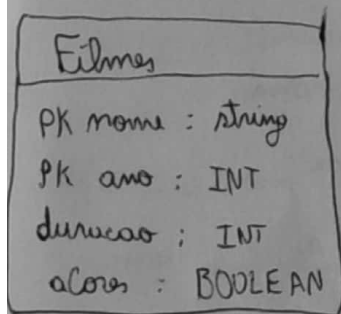
Eq. Filmagem (numero, nome Est.)

NOTA: Se tiver o atributo na classe de Suporte mesmo n sendo PK tb entra no schema relacional

• modelar a inscrição de um aluno a uma dada disciplina? Sendo que pode inscrever-se à mesma disciplina várias vezes.



• Conversão de UML para o modelo relacional



⇒ Filmes (nome, ano, duracao, cores)

- Uma relação é como se fosse uma tabela com linhas e colunas; Uma relação não pode ter linhas repetidas.

• Dependências Funcionais

Filmes (nome, ano, duracao, nome Est., morada Est.)

nome	ano	duracao	nome Est.	morada Estúdio
Star Wars	1977	124	Fox	10 Elm St. Los Angeles
Empire Strikes Back	1980	143	Fox	10 Elm St. Los Angeles
Gone with the Wind	1939	181	Paramount	44 Pine St. Los Angeles
Leon King	1999	124	Disney	56 Oak St. Los Angeles
Return of the Jedi	1983	165	Fox	10 Elm St. Los Angeles
Pocahontas	1995	115	Disney	56 Oak St. Los Angeles

⇒ Por vezes o esquema relacional tem falhas ⇒ Normalização

Problemas:

1. Redundância: a morada de um estúdio repete várias vezes
2. Pode dar origem a anomalias em "updates" ex: atualizar a morada no 1º tuplo e não o fazer no 2º e 5º tuplo
3. Pode dar origem a anomalias em "deletes" ex: se apagarmos o filme "Gone with the Wind" o estúdio "Paramount" desaparece da BD

• Normalização

- Estudo de Normalização permite-nos decompor uma relação em várias relações de modo que não haja este tipo de problemas / anomalias.
- Mas antes, devemos saber o que é uma Dependência Funcional

• Dependências Funcionais

$$X \rightarrow Y$$

Nota: $X, Y, Z \Rightarrow$ representar conjuntos de atributos
 $A, B, C \Rightarrow$ representar atributos individuais

Se fixarmos valores para os atributos X , os valores de Y não são únicos (não há repetidos)

• Chaves e Superchaves

- X é uma Superchave da Relação R se:

$X \rightarrow$ todos os atributos de R

X é uma chave da relação R , se nenhum subconjunto próprio de X for superchave de R

Nota: Y é subconjunto próprio de X , se:

$$Y \subseteq X \wedge Y \neq X$$

X é chave de $R \Rightarrow X$ é superchave de R

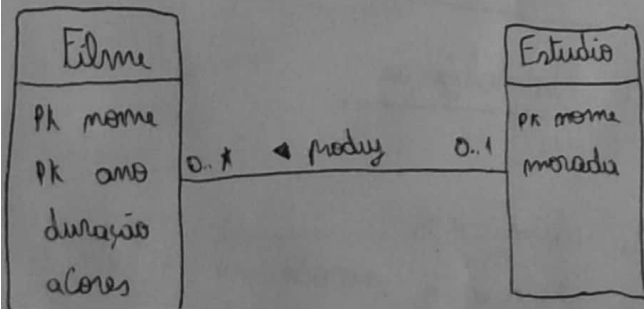
Regras sobre DF's

- Transitividade: Se $X \rightarrow Y$ e $Y \rightarrow Z$ então

$$X \rightarrow Z$$

Augmentation: Se $X \rightarrow Y$ então $XZ \rightarrow Y$

Exemplo: carros 1 e 2



Filmes (nome, ano, duração, cores, nome Est.)

nome ano \rightarrow duração cores

nome ano \rightarrow nome Estúdio

- Não pode haver 2 disciplinas que funcionem no mesmo dia da semana, à mesma hora e na mesma sala
- sala dia hora \rightarrow disciplina

de um conjunto de atributos: X^+
é o conjunto de todos os atributos que são
funcionalmente determinados por X .

Algoritmo para calcular X^+ :

1. $X^+ = X$
2. Se $A_1, A_2, \dots, A_m \rightarrow B$, e todos os A_i 's estão
contidos em X^+ , adicionar B a X^+
3. Termina quando não conseguirmos adicionar
mais nada a X^+

• Projeção de DF's

• Projeção = eliminar algumas colunas

Exemplo:

$R(ABCD)$ com DF's: $AB \rightarrow C, C \rightarrow D, D \rightarrow A$

Decompor em $R_1(ABC)$ e $R_2(AD)$

Quais as DF's em R_1 ?

$AB \rightarrow C$, mas também $C \rightarrow A$

Dada uma relação R com um conjunto de
DF's projetar R em R_1 . Quais as DF's de R_1 ?

1. Começar com as DF's de R e encontrar todas
as DF's não triviais que se podem deduzir a partir delas.
2. Ficar apenas com as DF's, cujos atributos fazem
parte do esquema de R_1

• Algoritmo para projetar as DF's de uma
relação R numa relação R_1

1. Seja T o eventual output. Inicialmente $T = \emptyset$
2. Para cada subconjunto X de atributos de R_1 ,
calcular X^+ (usando as DF's de R).
3. Inserir $X \rightarrow A$ em T , para todo o $A \in (X^+ - X)$,
desde que A seja atributo de R_1

Exemplo: Dada a relação $R(ABCDEF)$
com as seguintes DF's:

$AB \rightarrow C$

$BC \rightarrow A$

$DC \rightarrow D$

$D \rightarrow E$

$CF \rightarrow B$

$(AB)^+ = ?$



$(AB)^+ = AB$

$= ABC \quad \text{||} \quad \text{pois } AB \rightarrow C$

$= ABCD \quad \text{||} \quad \text{pois } BC \rightarrow D$

$= ABCDE \quad \text{||} \quad \text{pois } D \rightarrow E$

• Simplificação

1. Não é necessário calcular o fecho do
conjunto vazio, nem do conjunto de todos os at.
2. Se $X^+ =$ todos os atributos, não é
necessário calcular o fecho de subconjuntos
que contenham X

Exemplo: Seja $R(ABCD)$ com DF's:

$A \rightarrow B, B \rightarrow C, C \rightarrow D$

• Projetar R em $R_1(ACD)$ (eliminar B)

Fechos	T
$A^+ = ABCD$	$A \rightarrow C, A \rightarrow D,$
$C^+ = CD$	$C \rightarrow D$
$D^+ = D$	
$(CD)^+ = CD$	

• T é o conjunto de DF's de R_1 .

Mas podemos eliminar $A \rightarrow D$ porque
pode ser deduzida a partir das outras 2

• DF's de R_1 : $A \rightarrow C, C \rightarrow D$

5.

• Equivalência de conjuntos de DF's

- 2 conjuntos de DF's S e T não equivalentes, se o conjunto de instâncias que satisfaz S , for o mesmo que satisfaz T .

• Do exemplo anterior:

$\{A \rightarrow C, A \rightarrow D, C \rightarrow D\}$ é equivalente $\{A \rightarrow C, C \rightarrow D\}$

• BCNF

Uma relação R está em BCNF se:

- para todas as DF's de R não triviais $X \rightarrow Y$, X for superchave de R

• Decomposição de relações para BCNF

A ideia é usar uma DF $X \rightarrow Y$ que viola a condição de BCNF.

1. Calcular X^+ . (não pode ser todos os atributos)

2. Decompor R em 2 relações, R_1 e R_2 , com os seguintes esquemas:

① $R_1 \rightarrow X^+$

② $R_2 \rightarrow X \cup (\text{atributos}(R) - (X^+))$

• Base mínima de DF's

- Seja S um conjunto de DF's para uma rel. R
- Qualquer conjunto de DF's equivalente a S é uma base para S
- Uma base mínima para uma relação R , é uma base B que satisfaça:

① Todos as DF's em B , têm no lado direito 1 atributo

② Se removermos uma DF's de B , já não temos base

③ Se removermos um atributo do lado esquerdo de uma das DF's de B , deixa-se de ter uma base

• 3ª Forma Normal

Uma relação R está na 3FN se:
para todas as DF's não triviais $X \rightarrow Y$, se verificar que:

- X é superchave de R ou
- Y é membro de uma chave d R
- Se R está em BCNF \Rightarrow está em 3FN