

Modelação Conceptual de Base de Dados

Fernando Lobo

Base de Dados, Universidade do Algarve

1 / 56

Passos para criar uma base de dados

- 1 Compreender o problema no mundo real.
- 2 Especificá-lo usando um modelo conceptual.
- 3 Traduzir o modelo para um SGBD.
- 4 Criar esquema da BD usando uma “Data Definition Language” (DDL)
- 5 Carregar os dados
- 6 Desenvolver aplicações

2 / 56

Passos para criar uma base de dados (cont.)

- Para o passo 2 pode usar-se o modelo Entidade-Associação, UML, ou outro modelo conceptual.
- Por vezes o passo 2 é omitido (mas não é boa prática) e passa-se directamente para o modelo de dados do SGBD.
- A passagem de 2 para 3 pode ser automatizada.

3 / 56

Modelos conceptuais

- Modelos conceptuais mais usuais para BD:
 - ▶ Modelo Entidade-Associação (E/A)
 - ▶ UML
- Iremos aprender apenas modelação com UML

4 / 56

Unified Modeling Language (UML)

- UML é usado essencialmente para modelação de software com uma abordagem orientada a objectos.
- Um subconjunto do UML pode ser usado para modelação conceptual de base de dados.
- Tem uma notação gráfica.

5 / 56

Diagrama de classe

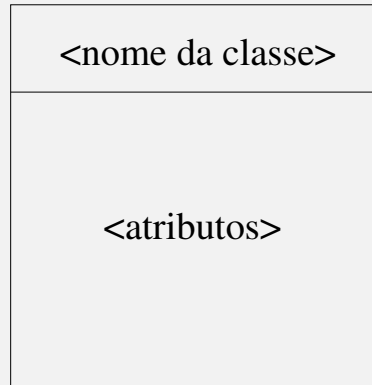
- Notação para descrever atributos (propriedades) e comportamentos (métodos) de objectos de uma classe.



6 / 56

Diagrama de classe

- Vamos ignorar os métodos nesta disciplina.



7 / 56

Exemplo

- Imaginem que têm de fazer uma BD para os Serviços Académicos.
- Vamos ter de guardar informação sobre os alunos.
- Para cada aluno podemos querer guardar o seu número de aluno, nome, telefone, morada, etc.
- Tudo isso são atributos/propriedades de um aluno.

8 / 56

Exemplo

- Faz sentido termos uma classe de alunos.
- Pensem numa classe como sendo um tipo de dados (uma estrutura em C)

Aluno
numero: int nome: string telefone: string morada: string

Exemplo

- Numa fase inicial, podemos omitir o tipo de dados dos atributos.

Aluno
numero nome telefone morada

Chave primária

- Conjunto de atributos que determina de forma única um objecto de uma classe.
- Especifica-se usando PK junto do(s) atributo(s).

Aluno
PK numero nome telefone morada

11 / 56

Conjunto de objectos de uma classe

- A futura BD terá um conjunto de objectos da classe Aluno.
- Esse conjunto pode ser visualizado na forma de uma tabela.
- Cada linha da tabela é um objecto da classe Aluno.

numero	nome	morada	telefone
34567	José Almeida	Rua da Prata, 27, Faro	96-3334598
45301	Maria Tavares	Av. da Liberdade, 13, Faro	91-9837788
38750	Paula Soares	Av. de Paris, 14, Olhão	91-4201314
...

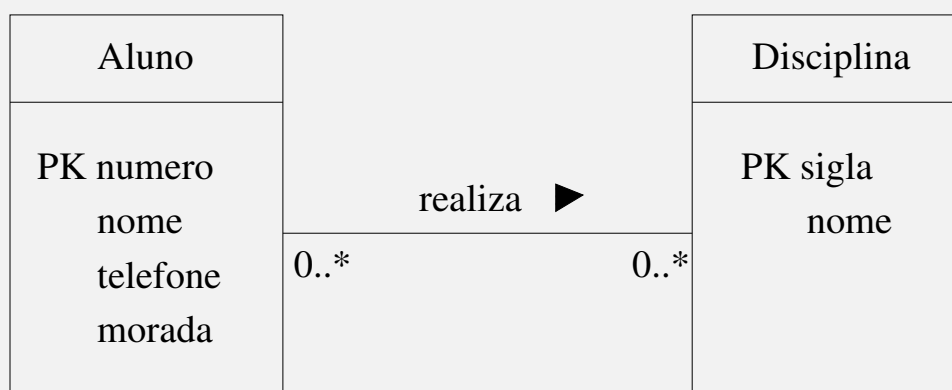
12 / 56

Associações

- Podemos ter associações entre objectos de classes diferentes.
- A associação é representada por uma linha com um nome.
- Exemplo: Podemos querer saber quais as disciplinas que cada aluno já fez.
 - ▶ Criamos uma classe Disciplina com atributos sigla e nome.
 - ▶ Unimos Aluno e Disciplina com uma linha e damos um nome que reflecte o tipo de ligação que existe.

13 / 56

Exemplo



- A seta facilita a leitura em português corrente.
- Um aluno realiza várias (0 ou mais) disciplinas.
- Uma disciplina é realizada por vários (0 ou mais) alunos.

14 / 56

Associações podem ser vistas em forma de tabela

Alunos

numero	nome	morada	telefone
34567	José Almeida	Rua da Prata, 27, Faro	96-3334598
45301	Maria Tavares	Av. da Liberdade, 13, Faro	91-9837788
38750	Paula Soares	Av. de Paris, 14, Olhão	91-4201314
...

Disciplinas

sigla	nome
BD	Bases de Dados
POO	Programação Orientada a Objectos
...	...

Realiza

aluno	disciplina
34567	BD
34567	POO
38750	POO
...	...

15 / 56

Multiplicidade de associações

- Multiplicidade é especificada nas extremidades da linha.
 - ▶ $m..n$ significa que 1 objecto do outro extremo está associado com um mínimo de m e um máximo de n objectos deste extremo.
 - ▶ $*$ → vários (ex: $1..*$ significa “pelo menos um”)

16 / 56

Exemplo



- Um aluno realiza um mínimo de 0 e um máximo de 10 disciplinas.
- Uma disciplina é realizada por um mínimo de 5 e um máximo de 30 alunos.

Multiplicidade de associações

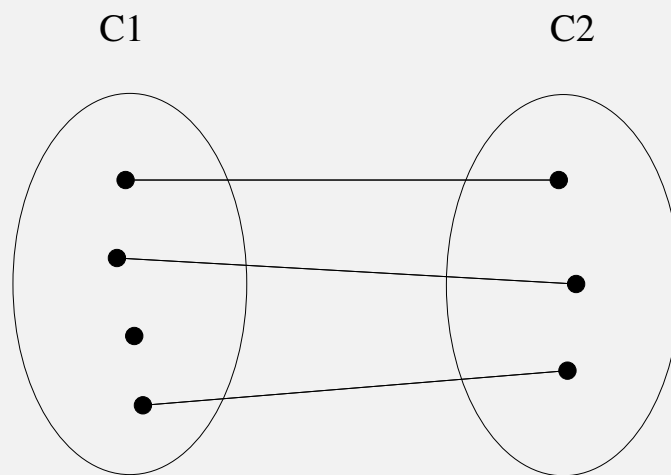
Dadas duas classes, C1 e C2,

um-um: cada objecto de C1 está associado no máximo a um objecto de C2 e vice-versa.

muitos-um: cada objecto de C1 está associado no máximo a um objecto de C2. Mas um objecto de C2 pode estar associado a vários objectos de C1.

muitos-muitos: cada objecto de C1 pode estar associado a vários objectos de C2 e vice-versa.

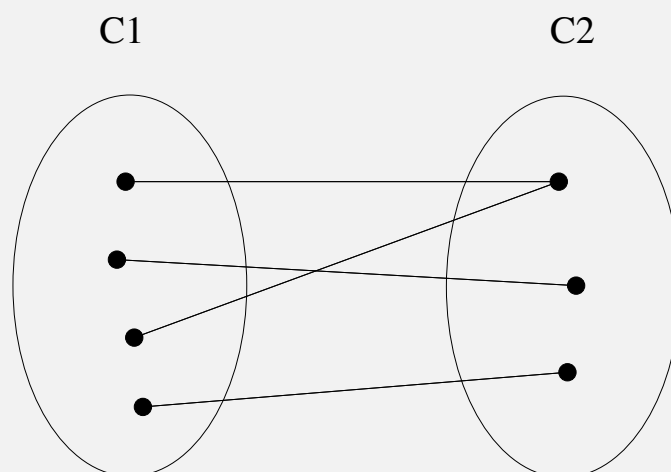
Multiplicidade um-um



- Cada objecto de C1 está associado no máximo a um objecto de C2 e vice-versa.

19 / 56

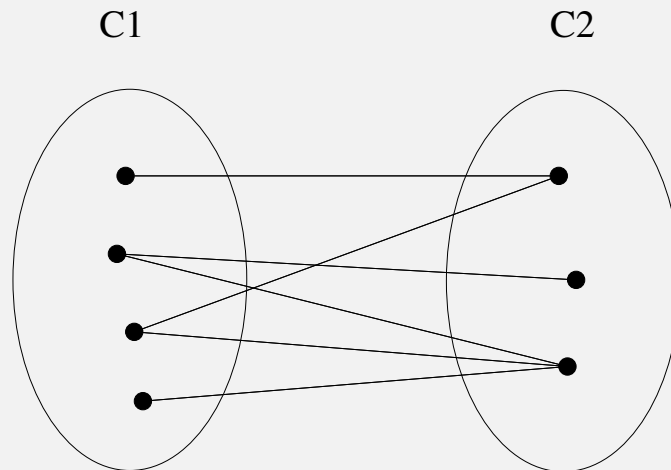
Multiplicidade muitos-um



- Cada objecto de C1 está associado no máximo a um objecto de C2.

20 / 56

Multiplicidade muitos-muitos



- Cada objecto de C1 pode estar associado a vários objectos de C2 e vice-versa. Mas um objecto de C2 pode estar associado a vários objectos de C1.

21 / 56

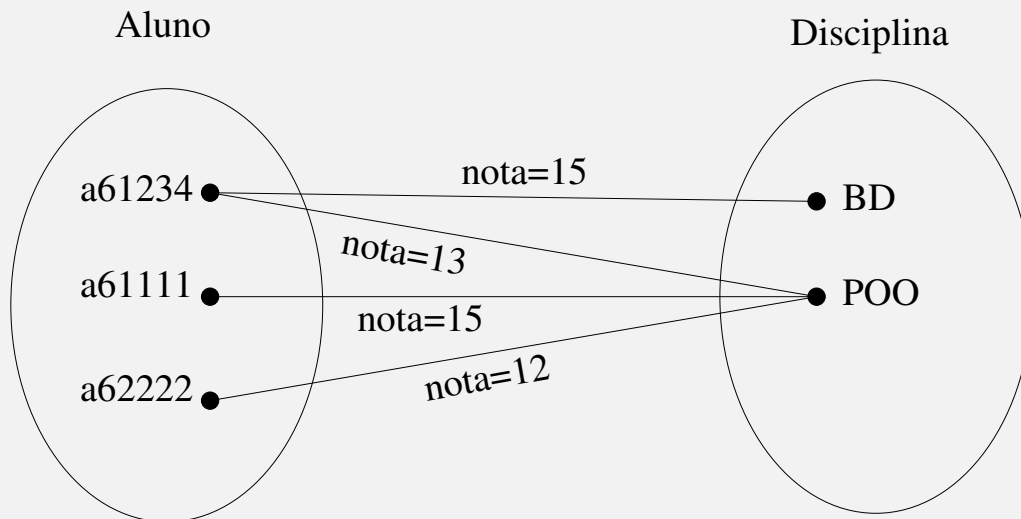
Classes associativas

- É permitido atributos em associações.
- Associação passa a ser uma “Classe Associativa”.

22 / 56

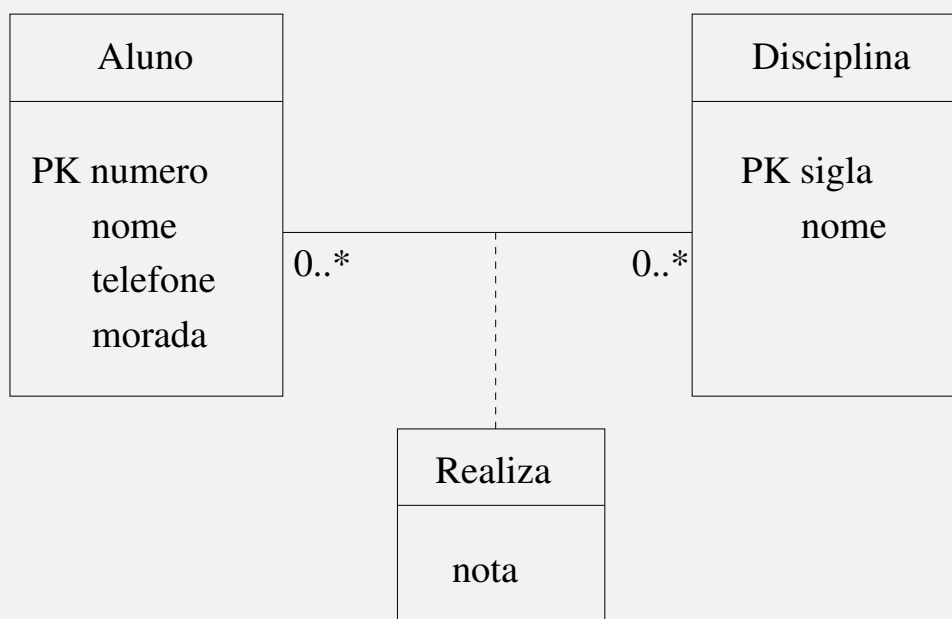
Exemplo

- Um aluno realiza uma disciplina e obtém uma nota.
- A nota é uma propriedade da associação aluno–disciplina.



23 / 56

Classe associativa em UML



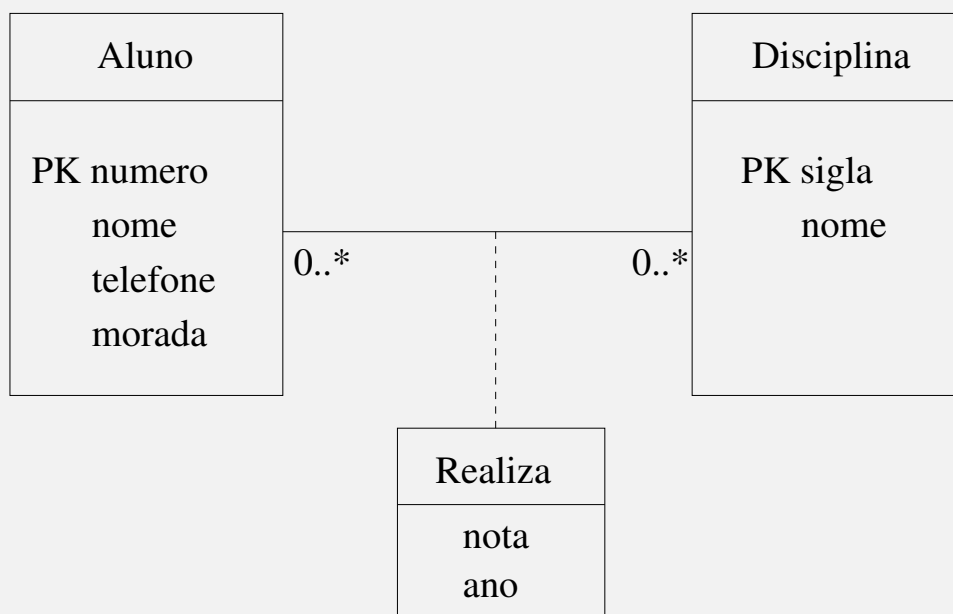
24 / 56

É necessário tomar decisões na modelação

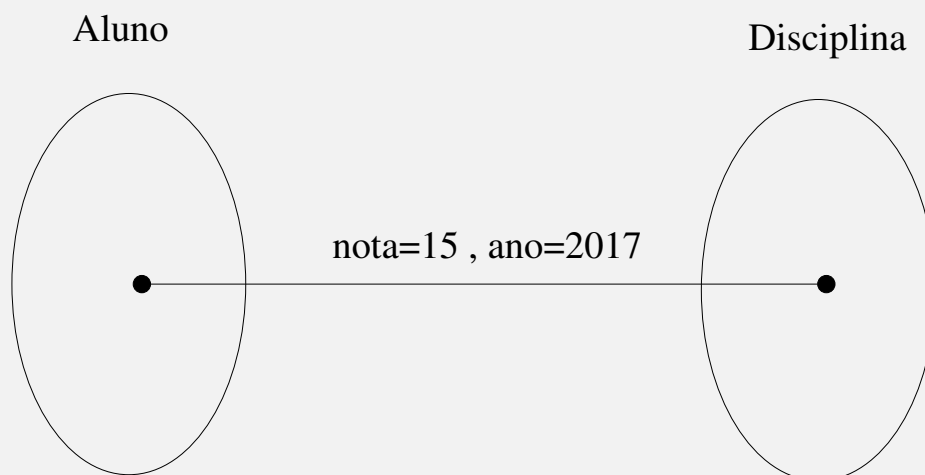
- E se quisermos saber o ano em que o aluno realizou a disciplina?
- Onde colocar o atributo **ano**?
- Devemos considerar BD 2017 = BD 2018 ?

25 / 56

- Se BD 2017 = BD 2018 então o ano pode ficar atributo de realiza.

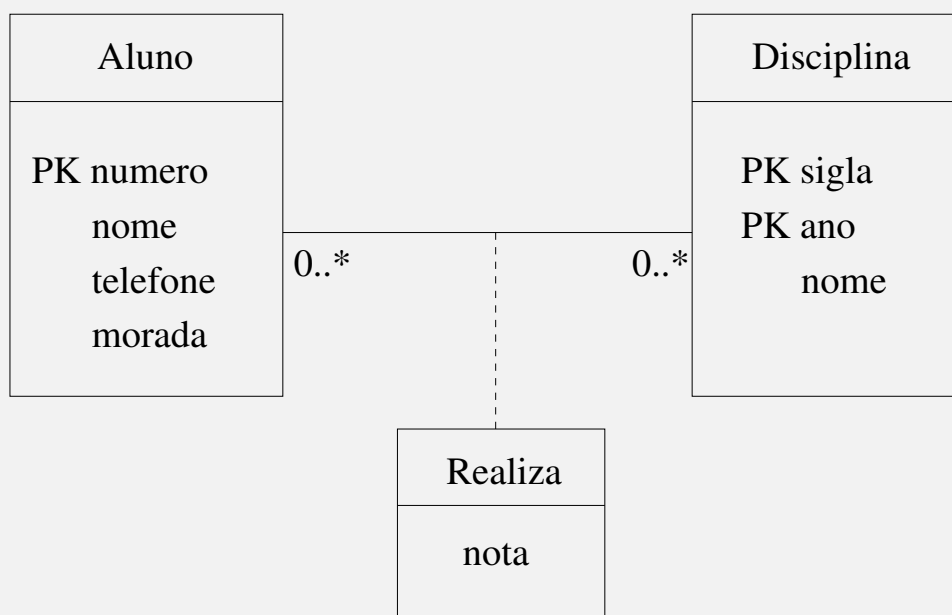


26 / 56



27 / 56

- Se BD 2017 \neq BD 2018, sigla não pode ser chave de Disciplina.
- Necessitaríamos de ter uma chave composta, ex: {sigla,ano}. Ou então um código de disciplina.



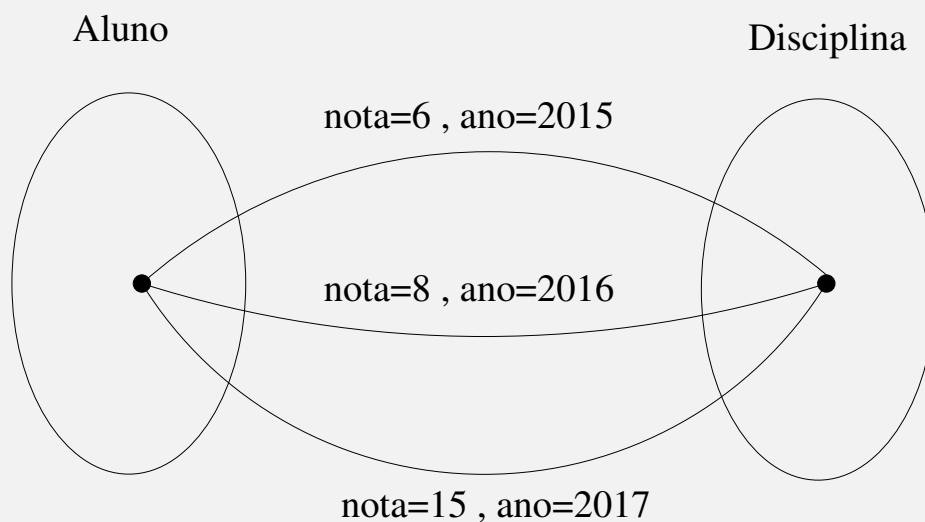
28 / 56

Mais decisões...

- Estamos a assumir que “Realiza” significa “ter aprovação à disciplina”.
- É razoável assumir que tal só acontece uma vez.
- E se quisermos modelar “inscrição à disciplina”?
- É perfeitamente razoável um aluno inscrever-se à mesma disciplina várias vezes.

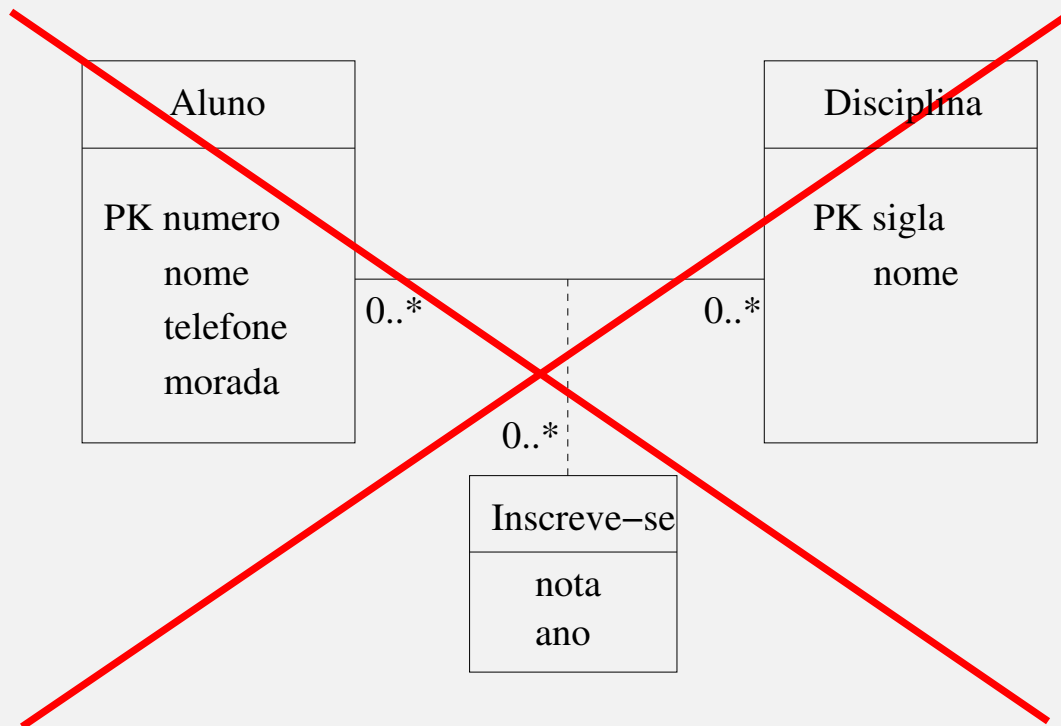
29 / 56

Conceptualmente ...



30 / 56

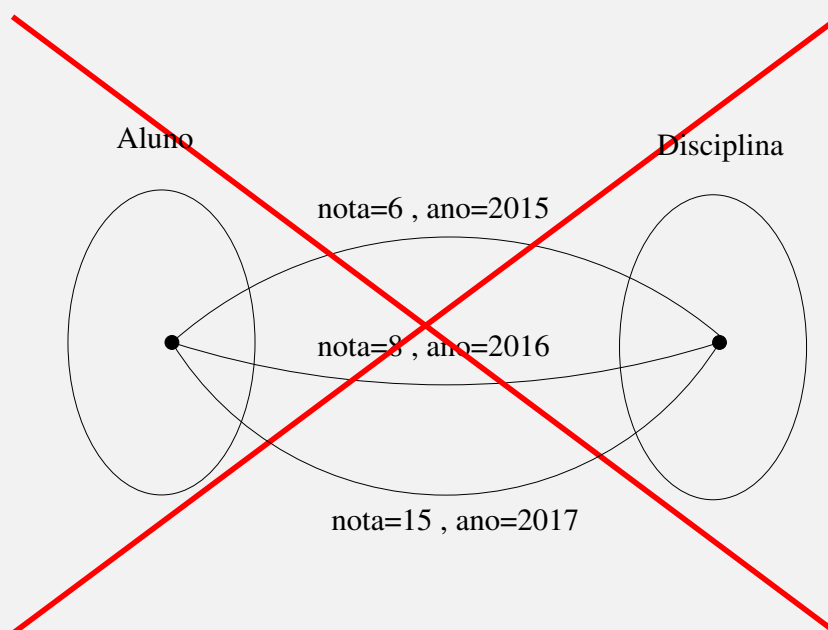
Erro comum em UML



31 / 56

Mas não se pode modelar assim

- Não é permitido ter mais do que uma associação com o mesmo nome entre dois objectos.
- i.e., não podemos ter várias associações de “inscrição” entre um determinado aluno e uma determinada disciplina.



32 / 56

- Mais adiante veremos como modelar este problema correctamente.
- Não esqueçam este exemplo. É um erro frequente. . .

Exemplo do vosso livro: filmes

- Um filme tem: nome, ano, duração, colorido ou preto/branco.
- Um actor/actriz tem: nome, morada, data de nascimento.
- Um estúdio tem: nome, morada.
- Um filme tem vários actores e um actor pode participar em vários filmes.
- Um filme pode ser produzido por um estúdio, um estúdio pode produzir vários filmes.

Conhecimento do mundo real

- Usa-se conhecimento do mundo real para modelar o problema.
- Não há uma única solução correcta, pode haver várias abordagens.
- Devemos assumir pressupostos razoáveis.

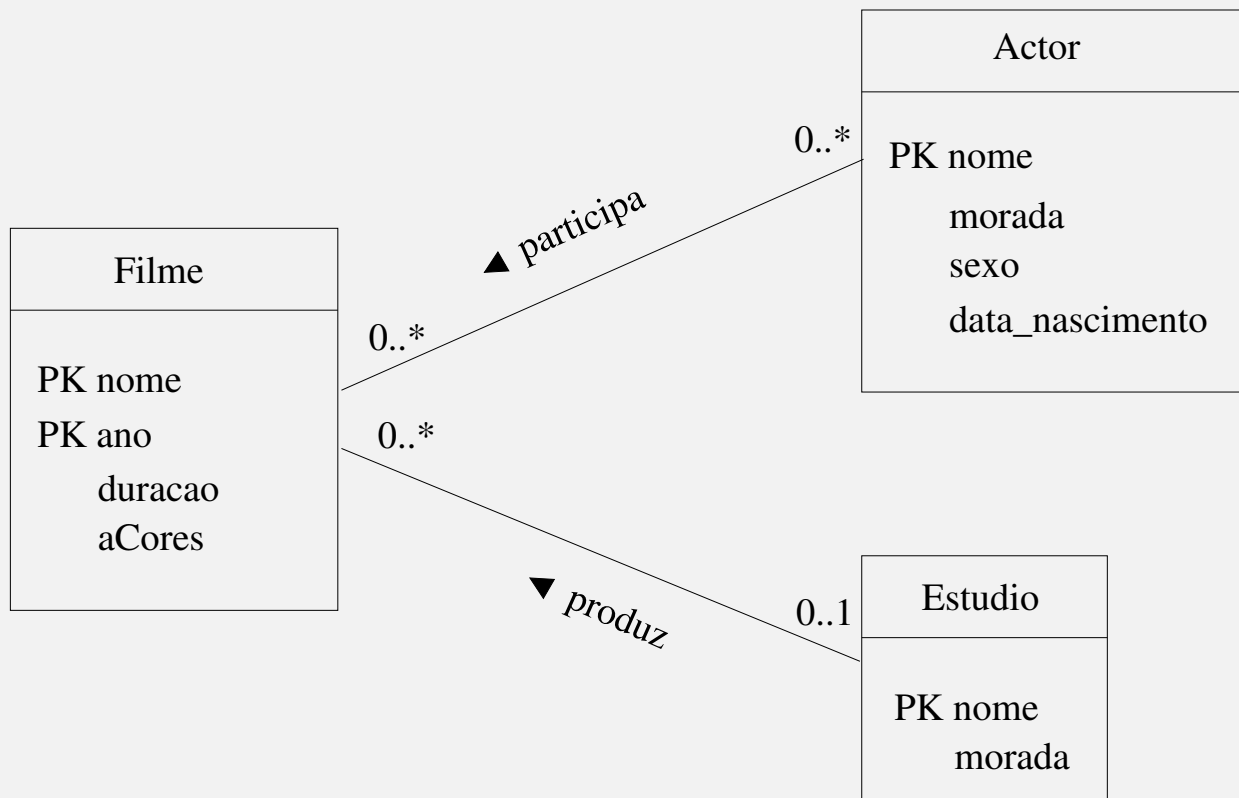
35 / 56

Pressupostos para a BD de filmes

- Não há dois actores com o mesmo nome.
- Não há dois estúdios com o mesmo nome.
- Não há dois filmes com o mesmo nome produzidos no mesmo ano (mas pode haver em anos distintos: Ex: Há o *King Kong* de 1933, mas também há o *King Kong* de 1976, e até ha o *King Kong* de 2005).
- NOTA: na prática o melhor seria colocar um atributo artificial — um código — para identificar univocamente cada filme, mas para já vamos ignorar isso.

36 / 56

Modelo UML



37 / 56

Sobre a chave primária composta

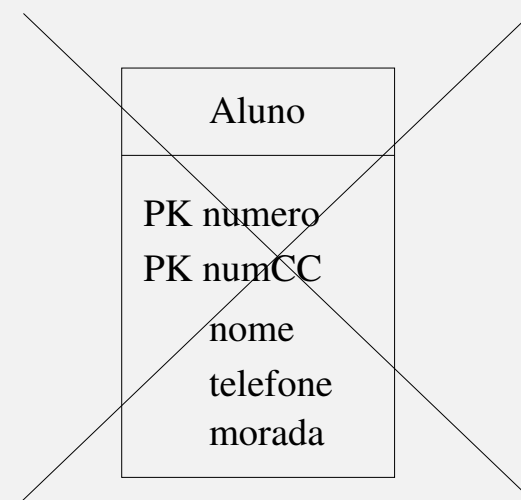
- No diagrama da classe Filme, colocamos PK em nome e PK em ano.
- Trata-se de uma chave composta. A chave primária é {nome,ano}.
- Nada impede que haja filmes com o mesmo nome.
- Nada impede que haja filmes produzidos no mesmo ano.
- O que estamos a proibir é que haja dois filmes com o mesmo nome produzidos no mesmo ano.

38 / 56

Várias alternativas para chave primária

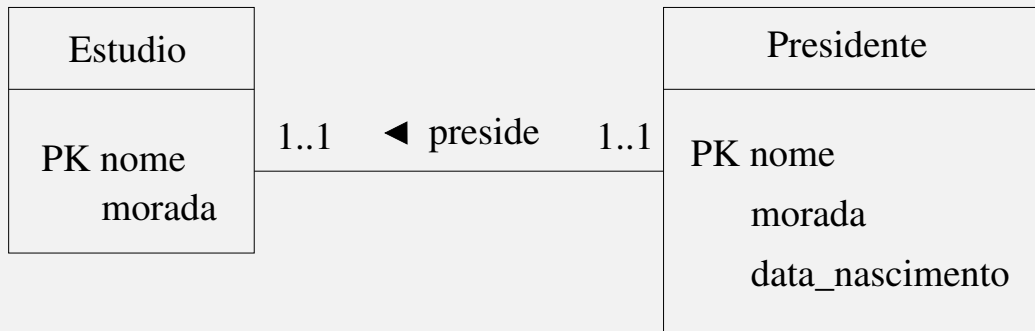
- Pode haver várias alternativas para chaves primárias. Nesse caso escolhemos uma delas como chave primária.
- Exemplo:
 - ▶ no caso da BD dos SA poderemos querer guardar o número do cartão de cidadão do(a) aluno(a).
 - ▶ temos duas opções para chave primária: número-de-aluno ou número-de-CC.

Um erro comum



Estúdio tem um presidente

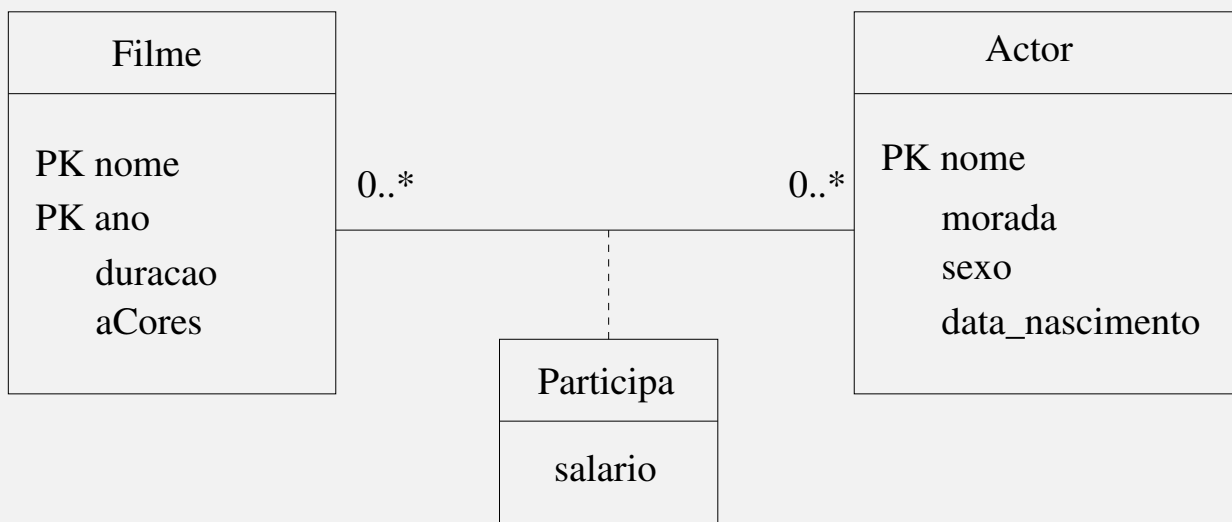
- Um estúdio tem um e um só presidente.
- Vamos assumir que um presidente só pode ser presidente de um estúdio.



41 / 56

Vencimento dos actores

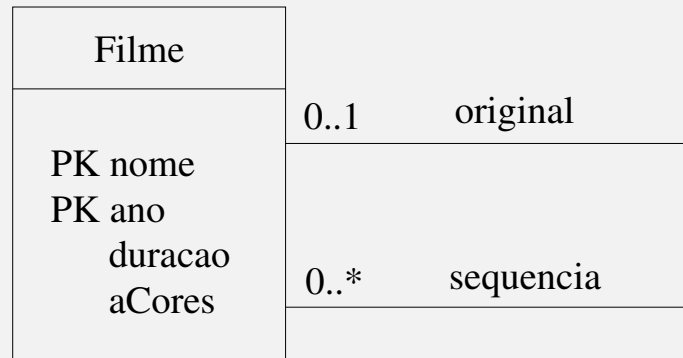
- Um actor recebe consoante o filme em que participa.
- Solução: Classe associativa.



42 / 56

Associações com a própria classe

- Pode haver associações entre objectos da mesma classe.
- Cada objecto representa um papel diferente na associação.
- Ex: um filme pode ser sequência de um outro filme.



43 / 56

Visualizando em forma de tabela

sequência	original
The Empire Strikes Back	Star Wars
The Return of the Jedi	The Empire Strikes Back
Rocky II	Rocky I
Rocky III	Rocky II
...	...

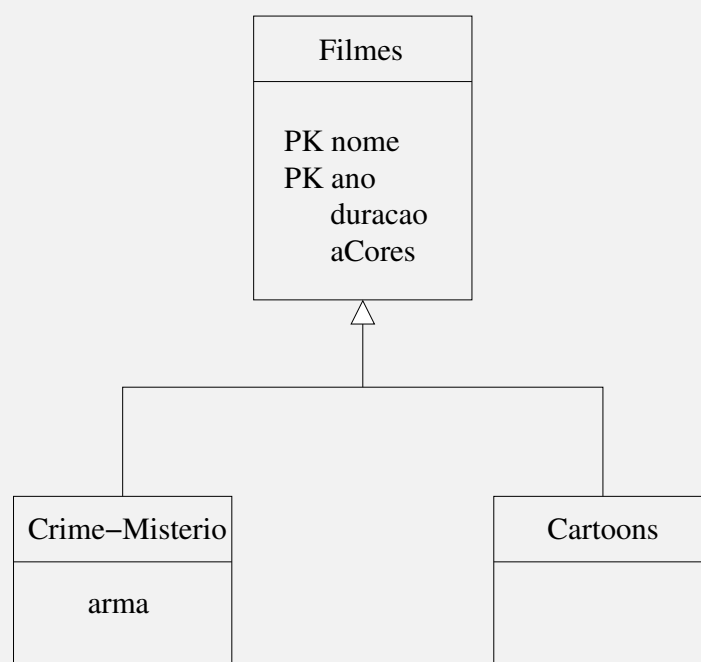
44 / 56

Subclasses

- Uma classe pode ter subclasses.
- Ex: Um Cartoon é um tipo de filme, Crime-Mistério também é um tipo de filme.
- Cartoons é uma subclasse de Filmes, Crime-Mistério é outra sub-classe de Filmes.
- Subclasse aponta para a superclasse com uma linha com um triângulo na extremidade.
- As subclasses podem ter atributos.
 - ▶ ex: arma do crime para os filmes de Crime-Mistério.

45 / 56

Subclasses



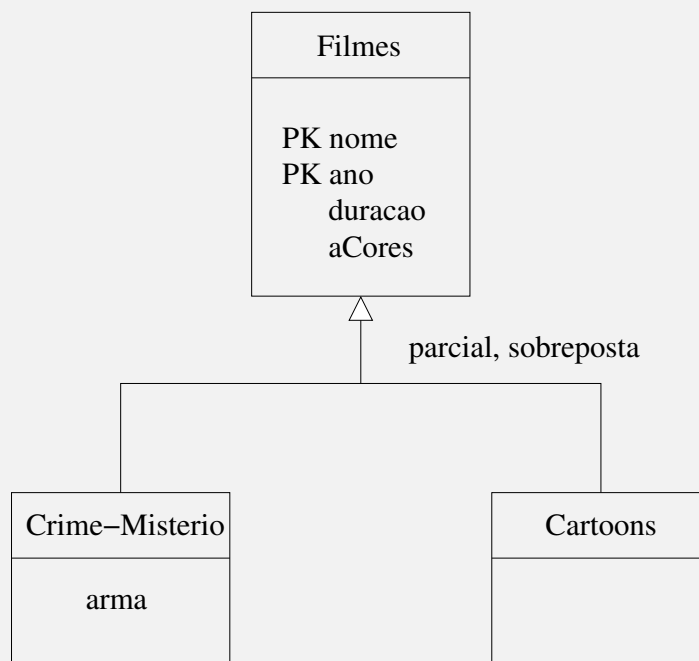
46 / 56

Subclasses

- As subclasses de uma classe podem ser:
 - ▶ completas ou parciais
 - ★ completa se cada objecto da superclasse está numa das subclasses, parcial caso contrário.
 - ▶ disjuntas ou sobrepostas
 - ★ disjunta se um objecto não pode estar em mais do que uma subclasse, sobreposta caso contrário.
 - ▶ É costume anotar a classificação no diagrama de forma textual

47 / 56

Subclasses de Filmes (parciais e sobrepostas)



- Porquê sobrepostas? Porque um filme pode ser simultaneamente um cartoon e um filme de crime-mistério (ex: *Who Framed Roger Rabbit?*)

48 / 56

Agregação

- Associação com o significado que objectos de um dos lados podem “ser possuídos” ou “fazer parte de” objectos do outro lado.
- Ex: um filme pode ser produzido por um estúdio.
 - ▶ (o estúdio é “dono do” filme)



representa-se por  no lado da classe que é "dona de"

 significa 0..1

Composição

- Semelhante à agregação mas a posse é obrigatória.
- Ex: um filme tem de ser forçosamente produzido por um estúdio.



representa-se por  no lado da classe que é "dona de"

 significa 1..1

Agregação e Composição

- Não é necessário dar um nome à associação.
- O nome está implícito pela “posse” dos objectos.

Classe de suporte

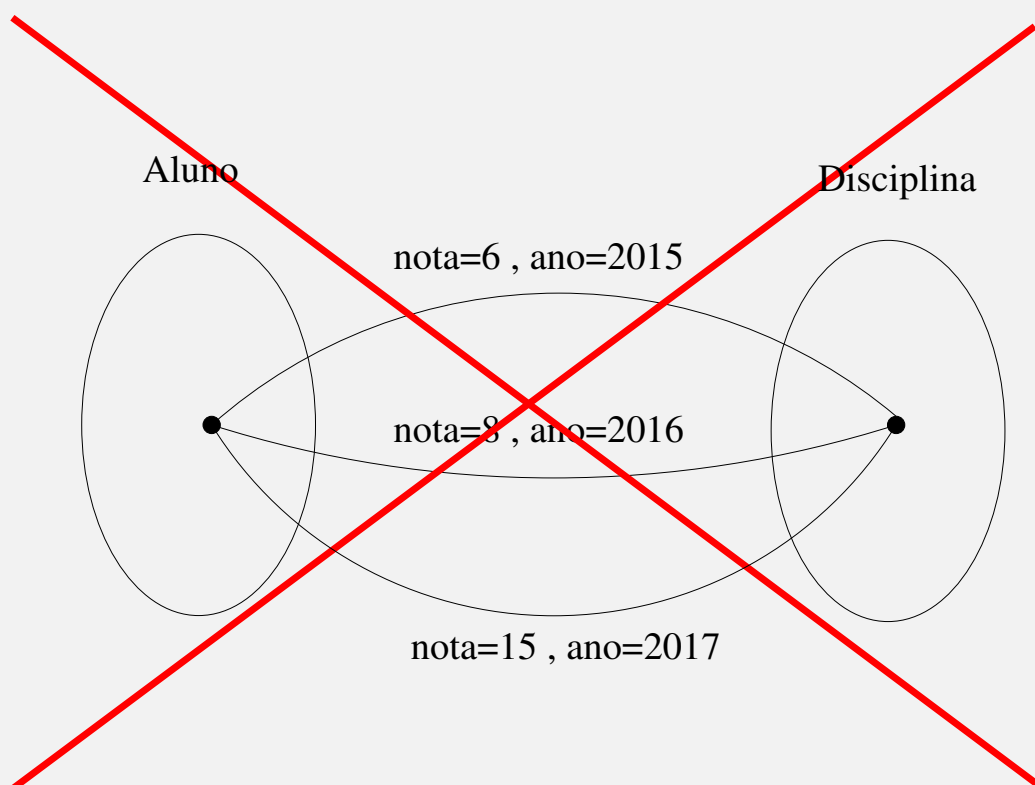
- Por vezes podemos ter uma classe que serve de suporte a outra classe.
- Objectos desta classe só fazem sentido no contexto de outra classe.
- Ex: Um estúdio pode ter várias equipas de filmagem: equipa 1, 2, 3, ...
- A equipa de filmagem é uma classe de suporte (ou classe fraca) a um estúdio.
- Modelamos este tipo de associação com composição e colocamos uma caixinha assinalada com PK no diagrama de classe de suporte.
- Uma equipa de filmagem só pode ser identificada se for “buscar” a chave do estúdio respectivo.

Classe de suporte



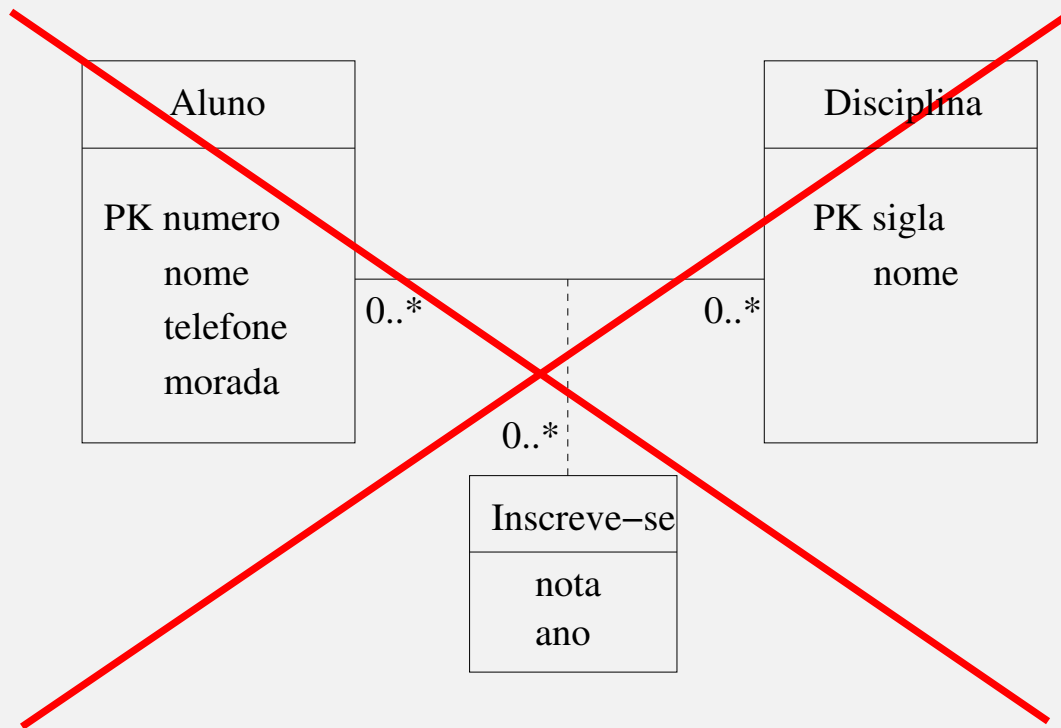
53 / 56

Voltando ao problema das inscrições de alunos



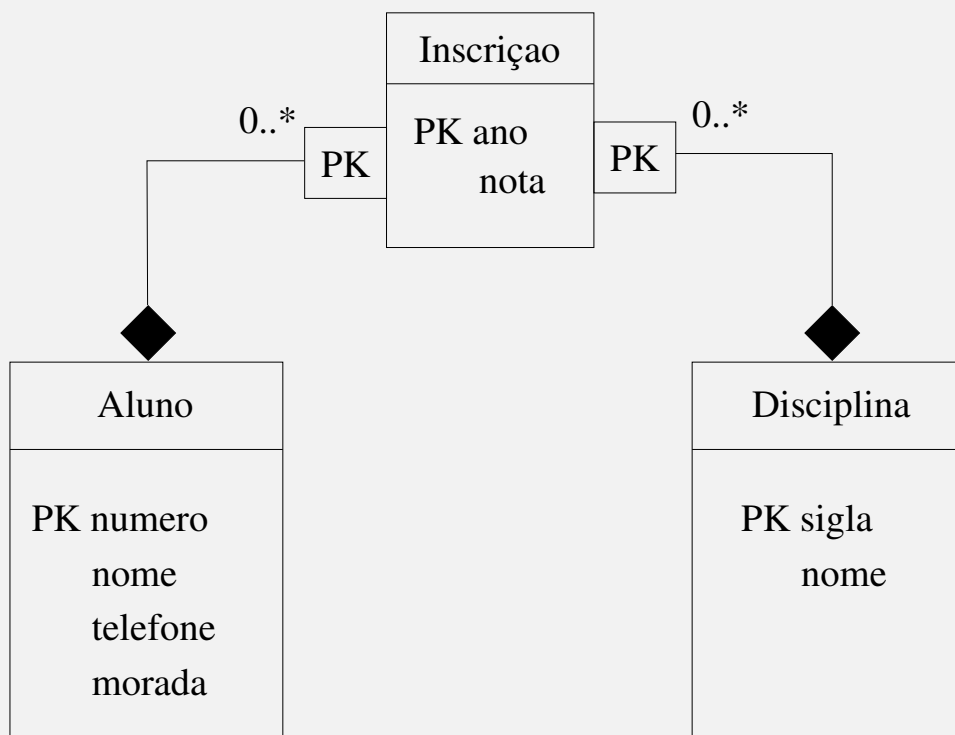
54 / 56

Assim NÃO



55 / 56

Assim SIM



56 / 56