Cadeira de Tecnologias de Informação

Ano lectivo 2010/11

Regras de Transformação, ou *Mapping,* do Diagramas de Classes para o Modelo Relacional

Tópicos

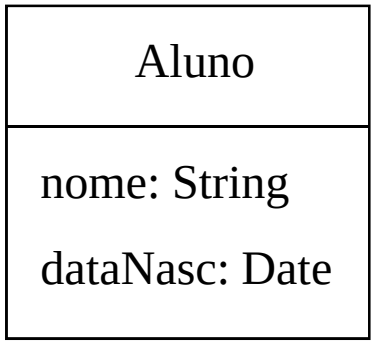
- **Conceitos**
- **Regras de Transformação**

Modelo Relacional vs Paradigma OO

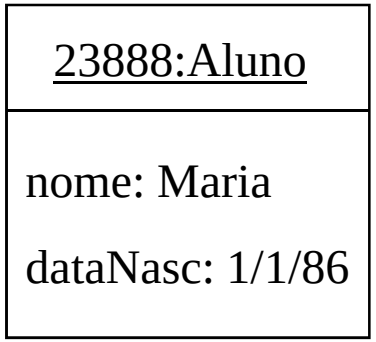
Tabela

alunoID	nome	morada	dataNasc
23888	Maria	567	01/01/86
23889	José	767	20/07/90
20123	Tiago	143	01/03/78
25978	Rosa	767	14/02/94
12870	Ana	231	07/08/64

Classe



Objecto



Classes e respectivos Relacionamentos como relações

Objectos e respectivos relacionamentos como linhas

Conceitos do Modelo Relacional

Resumo dos conceitos da estrutura de dados do modelo relacional:

Termo Formal Relacional	Definição
Esquema de Relação	Definição da Relação
Tabela	Significa o mesmo que Relação
Tuplo	Linha de uma tabela (instância)
Chave Primária	Atributo ou conjunto de atributos que identificam univocamente cada ocorrência da relação
Chave Estrangeira	Conjunto de um ou mais atributos que são chave numa outra relação
Atributo	Nome de um atributo de uma relação
Domínio de um atributo	Conjunto de valores possíveis do atributo

Conceito de "Regra de Transformação" para o Modelo Relacional

Conjunto de regras que regem a transição de classes, e respectivos atributos e Relacionamentos, do Diagrama de Classes para o Modelo Relacional, de forma a garantir que não exista perda de informação

Implicações das Regras de transformação na Normalização nas tabelas

Chave Primária – Todas a tabelas devem possuir uma chave primária e nenhum atributo da chave primária pode ter o valor *null*

Redundância Controlada – Alguns atributos aparecem repetidos, mas tal repetição verifica-se apenas em atributos que são chaves estrangeiras (primárias noutras relações)

Regras de Transformação

- **1ª Regra - Classes**

Uma classe com estrutura simples é

mapeada numa relação

E os objectos?

- Um objecto derivado de uma classe simples é mapeado num tuplo da relação correspondente

E os atributos?

- Os atributos definidos originalmente na classe são mapeados no esquema da relação
  - » O tipo de atributos é mapeado por aproximação (ex: "Integer" -> int, "String" -> varchar2(..))

Regras de Transformação

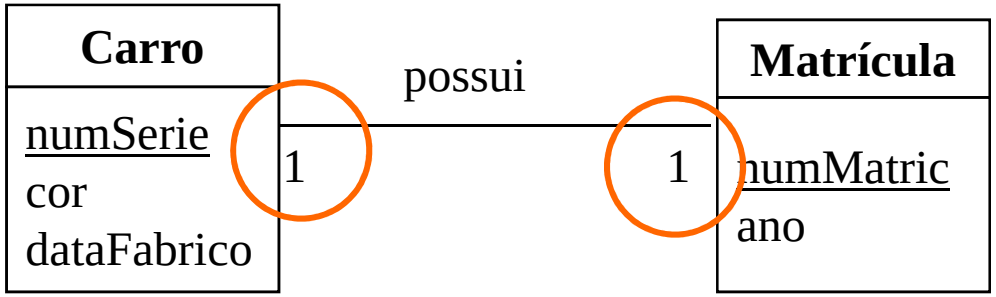
- **2ª Regra – Chaves Primárias**

Todas as tabelas devem possuir uma chave primária.

- Sempre que um ou mais atributos de uma classe apresentem o estereótipo "OID" (*object identifier*) podemos mapear directamente esse atributo como chave primária da tabela correspondente
- Quando nada é dito explicitamente, assume-se que existe na classe um ou mais atributos que identificam univocamente um tuplo da relação

Regras de Transformação

- **3ª Regra – Associação de "um para um" (1/3)**



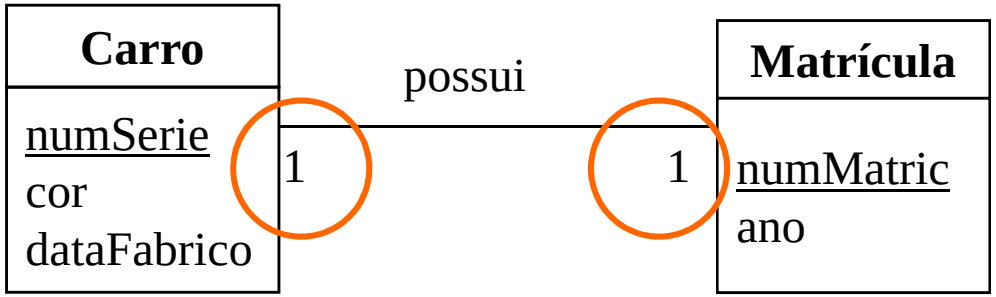
- Uma das tabelas deverá receber como chave estrangeira a chave primária da outra tabela.
  - » Carro (numSerie, cor, dataFabrico, *numMatric*)
  - » Matrícula (numMatric, ano)

⇨ **A chave primária é sempre representada em sublinhado e a chave estrangeira é representada em itálico ou em sublinhado interrompido.**

Recebe chave estrangeira a tabela que fizer mais sentido.

Regras de Transformação

- **3ª Regra – Associação de "um para um" (2/3)**

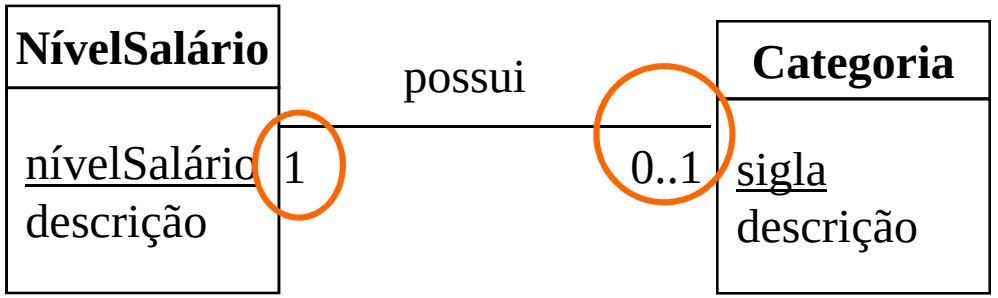


- Outra solução é agregar os atributos definidos nas classes originais num único esquema de relação, por motivos de simplificação ou optimização
  - » Carro (numSerie, cor, dataFabrico, numMatric, ano)

Apenas quando a semântica não apresenta grande distinção.

Regras de Transformação

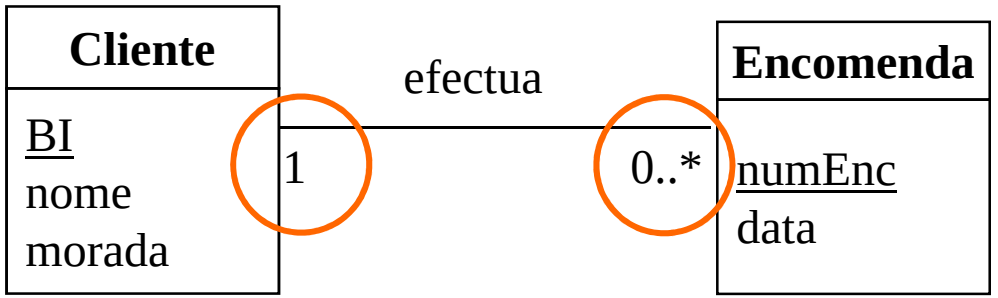
- **3ª Regra – Associação de "um para um"(3/3)**
  - » caso em que a multiplicidade é 0..1



- Atendendo a que a associação não é obrigatória de um dos lados, a chave estrangeira fica obrigatoriamente na tabela que corresponde ao lado 0..1
  - » Categoria (sigla, descrição, *nívelSalário*)
  - » NívelSalário (nívelSalário, descrição)

Regras de Transformação

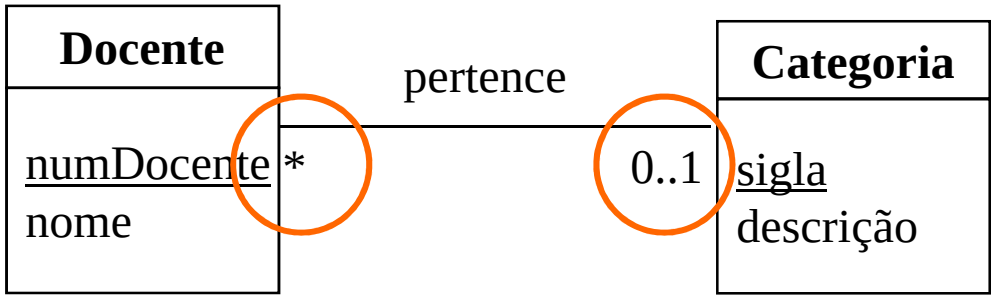
- **4ª Regra – Associação de "um para muitos"(1/2)**



- A tabela em que a "informação está repetida" é que recebe a chave estrangeira; ou seja, a parte dos "muitos" é que recebe a chave estrangeira
  - » Encomenda (numEnc, data, *BI*)
  - » Cliente(BI, nome, morada)

Regras de Transformação

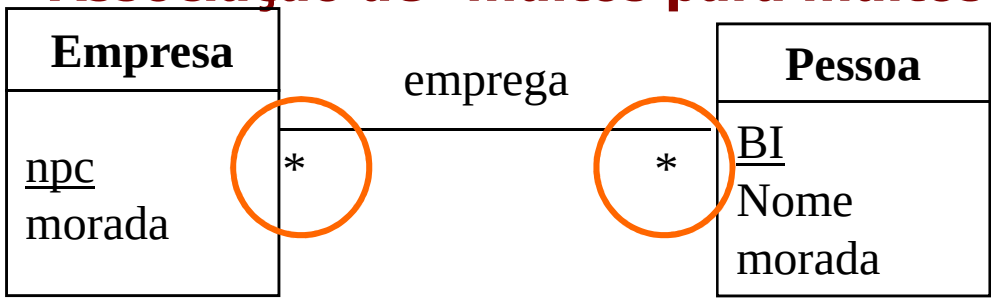
- **4ª Regra – Associação de "um para muitos"(2/2)**
  - » caso em que a multiplicidade é 0..1



- Este caso é idêntico à situação anterior, à excepção de que sigla pode ter o valor nulo, no caso do Docente não ter categoria:
  - » Categoria (sigla, descrição)
  - » Docente (numDocente, nome, *sigla*)

Regras de Transformação

- **5ª Regra – Associação de "muitos para muitos"**



- Este caso exige a introdução de uma nova tabela, cujos atributos são as chaves primárias das duas tabelas
  - » Empresa (npc, morada)
  - » Pessoa (BI, nome, morada)
  - » Emprega (*npc*, *BI*)

⇨ Chaves primárias e estrangeiras simultaneamente, são representadas em sublinhado e itálico ou em sublinhado e sublinhado interrompido.

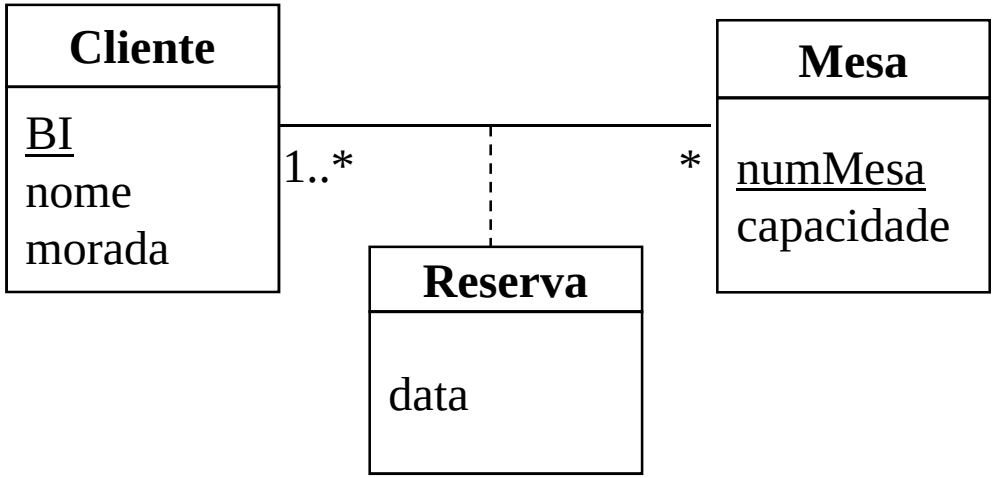
Regras de Transformação

- **6ª Regra – Classes associativas (1/3)**

Utiliza-se uma das regras correspondentes à associação, com a ressalva de que os atributos da classe associativa são recebidos pela tabela que recebe as chaves

Regras de Transformação

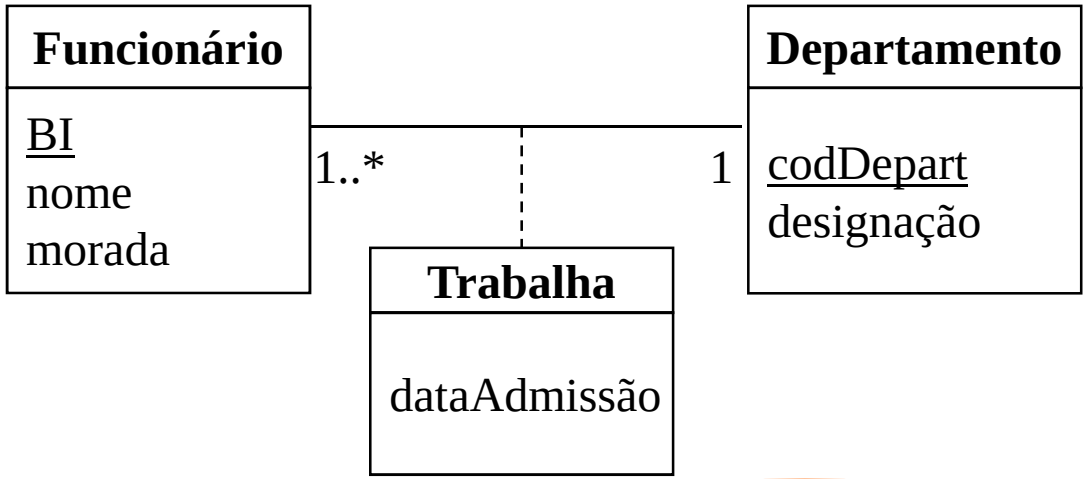
- **6ª Regra – Classes associativas (2/3)**
  - » caso em que a multiplicidade é "muitos para muitos"



  - » Cliente (BI, nome, morada)
  - » Mesa (numMesa, capacidade)
  - » Reserva (*BI, numMesa*, data)

Regras de Transformação

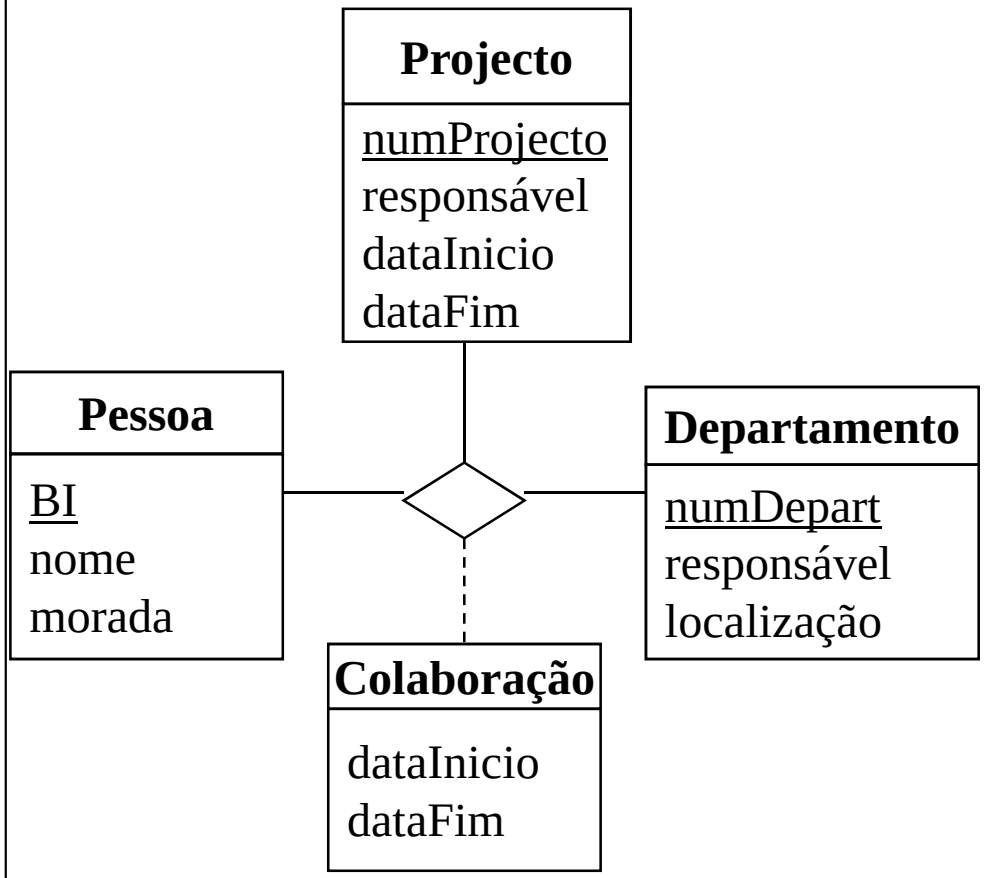
- **6ª Regra – Classes associativas (3/3)**
  - » caso em que a multiplicidade é "um para muitos"



  - » Funcionário (BI, nome, morada, *codDepart*, dataAdmissão)
  - » Departamento (codDepart, designação)

Regras de Transformação

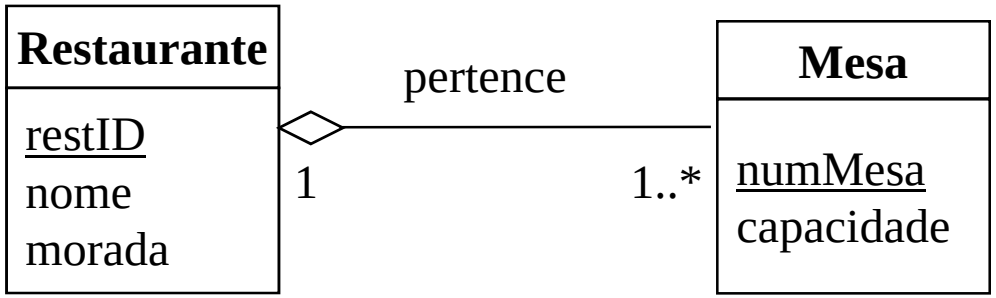
- **7ª Regra – Associações N-árias (N>=3)**
  - » Idêntico ao que acontece com as classes associações em associações binárias



- » Pessoa (BI, nome, morada)
- » Departamento (numDepart, responsável, localização)
- » Projecto (numProjecto, responsável, dataInicio,dataFim)
- » Colaboração (*BI*, *numProjecto*, *numDepart*, dataInicio, dataFim)

Regras de Transformação

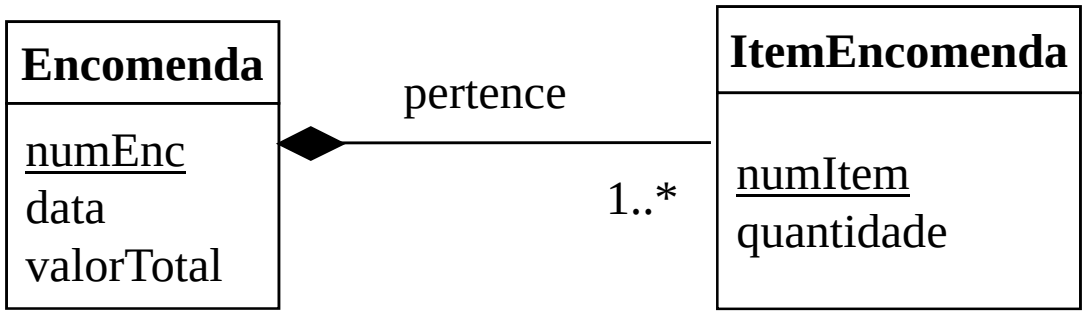
- **8ª Regra – Associação de Agregação**
  - » Exemplo:



- Utilizam-se as mesmas regras de transformação das associações com a mesma multiplicidade
  - » Restaurante (restID, nome, morada)
  - » Mesa (numMesa, capacidade, *restID*)

Regras de Transformação

- **9ª Regra – Associação de Composição**



- A tabela que equivale à classe de composição (partes) fica com a chave estrangeira. Esta chave também fará parte da sua chave primária.
  - » Encomenda (numEnc, data, valorTotal)
  - » ItemEncomenda (numItem, *numEnc*, quantidade)

Regras de Transformação

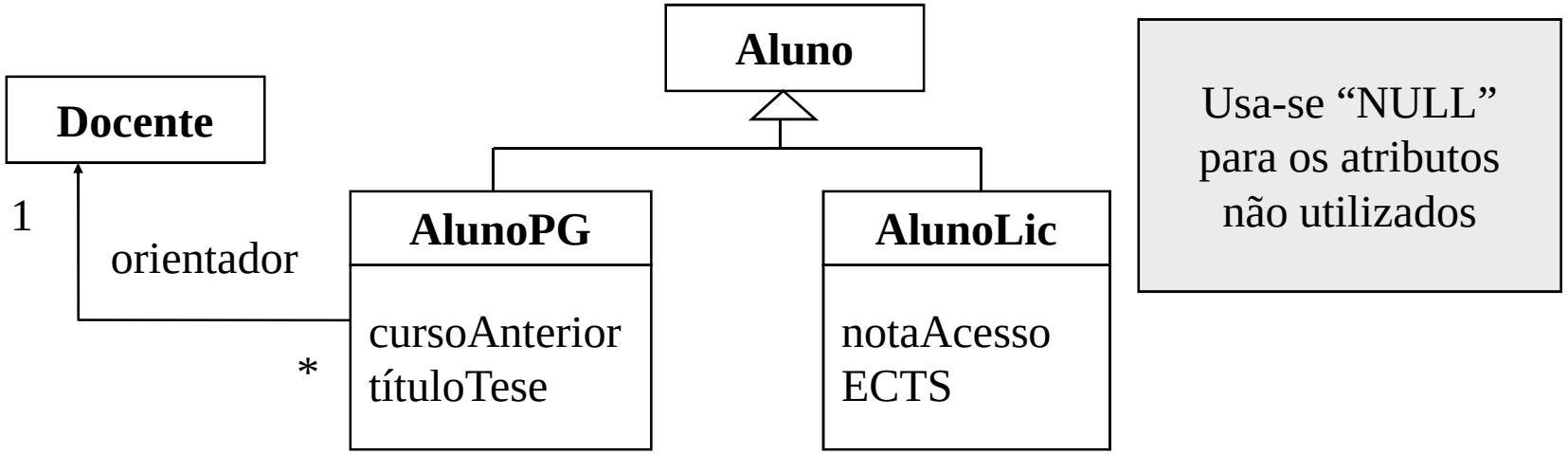
- **10ª Regra – Associação de Generalização (1/5)**

Esta transposição varia consoante a natureza da identidade das subclasses, resumindo-se a 3 soluções:

1) Esmagamento das classes da hierarquia num único esquema relacional correspondente à superclasse
2) Considerar apenas esquemas correspondentes às subclasses
3) Considerar todas as classes da hierarquia

Regras de Transformação

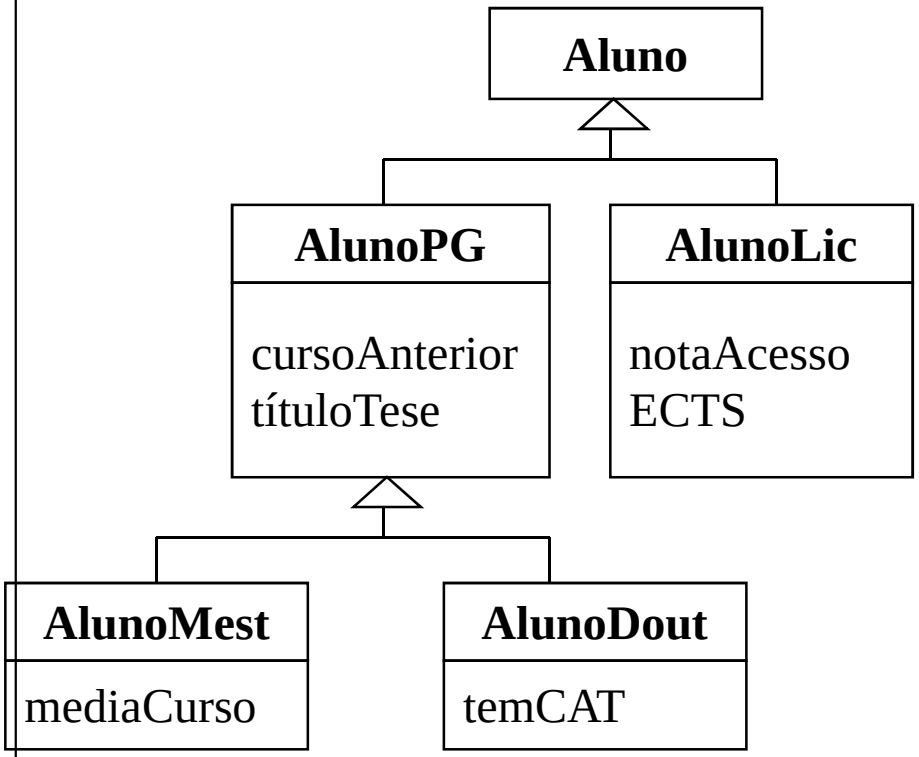
- **10ª Regra – Associação de Generalização (2/5)**
  - » Esmagamento das classes da hierarquia num único esquema correspondente à superclasse



- Esta solução é mais adequada quando as subclasses não diferem muito em termos de estrutura ou não apresentam uma semântica forte ao nível da sua própria identificação.
  - » Aluno (alunoID, nome, morada, notaAcesso, ECTS, cursoAnterior, títuloTese, *orientador*)

Regras de Transformação

- **10ª Regra – Associação de Generalização (3/5)**
  - » Considerar apenas esquemas correspondentes às subclasses "folhas" (subclasses não generalizadas)

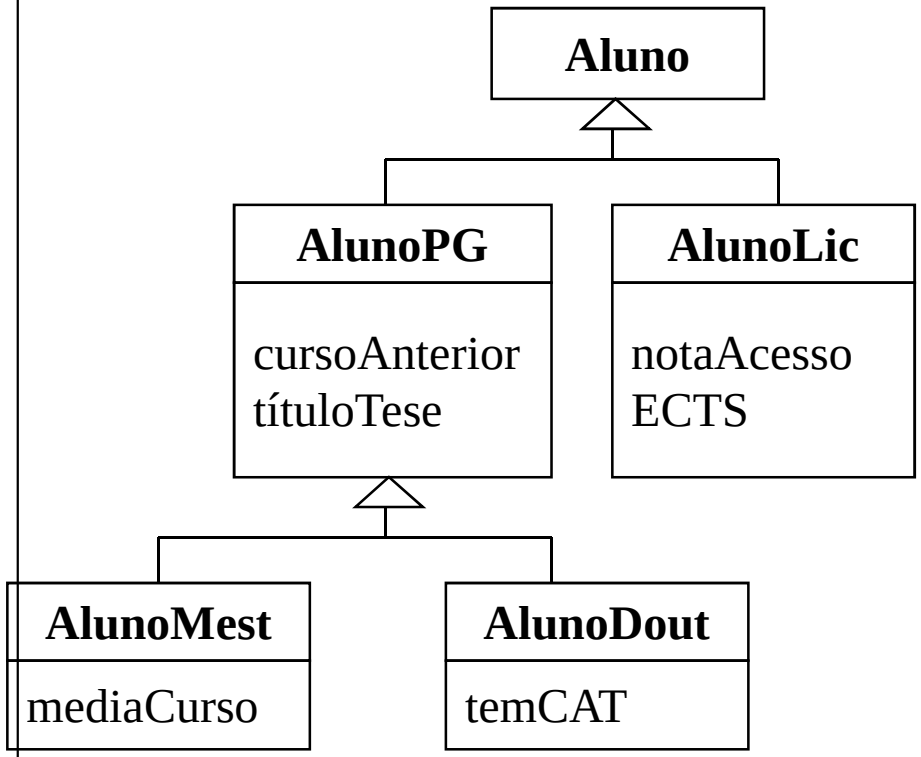


- Todos os atributos da superclasse são reproduzidos (duplicados) em todos os esquemas correspondentes às subclasses.

Apenas funciona se a superclasse não tiver instâncias (for abstracta)

Regras de Transformação

- **10ª Regra – Associação de Generalização (3/5)**
  - » Considerar apenas esquemas correspondentes às subclasses "folhas" (subclasses não generalizadas)



» AlunoLic (alunoLicID, nome, morada, dtNasc, notaAcesso, ECTS)

» AlunoMest (alunoMestID, nome, morada, dtNasc, cursoAnterior, títuloTese, *orientador*, médiaCurso)

» AlunoDout (alunoDoutID, nome, morada, dtNasc, cursoAnterior, títuloTese, *orientador*, temCAT)

Apenas funciona se a superclasse não tiver instâncias (for abstracta)

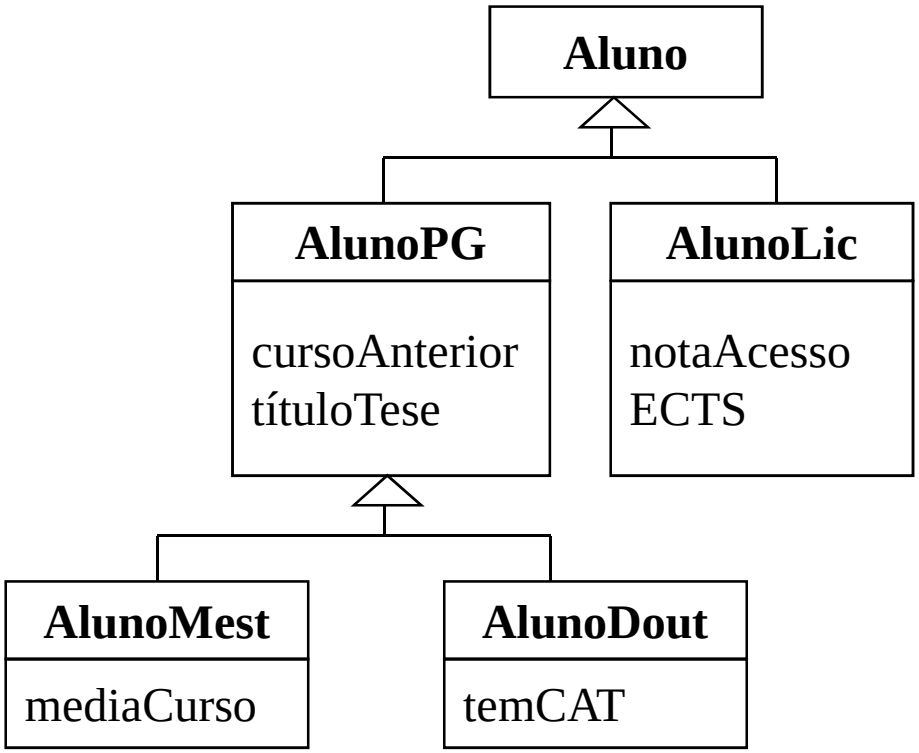
Regras de Transformação

- **10ª Regra – Associação de Generalização (4/5)**
  - » Considerar todas as classes da hierarquia

- Solução de considerar todos os esquemas correspondentes a todas as subclasses e superclasses
- Necessita da existência de uma malha de esquemas ligados e mantidos à custa de regras de integridade referencial
- Evita a duplicação de informação, mas sugere pulverização da informação entre diferentes esquemas, o que pode trazer problemas de desempenho

Regras de Transformação

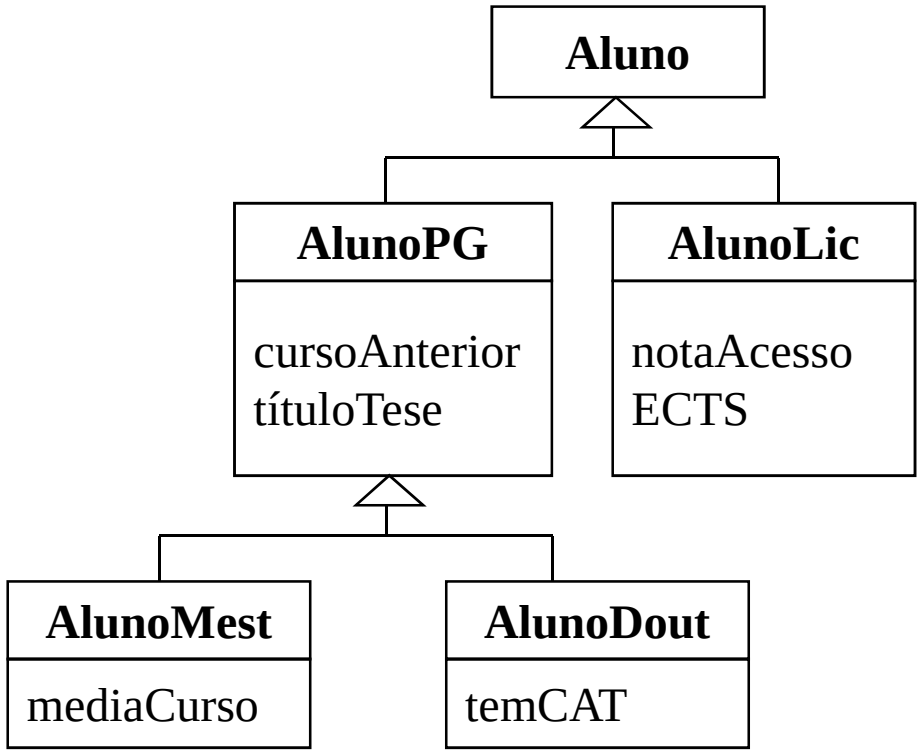
- **10ª Regra – Associação de Generalização (5/5)**
  - » Considerar todas as classes da hierarquia. Neste caso todos os alunos são identificados por alunoID



- » Aluno (alunoID, nome, morada, dtNasc)
- » AlunoLic (alunoID, notaAcesso, nrDisciplinas, ECTS)
- » AlunoPG (alunoID, cursoAnterior, títuloTese, *orientador*)
- » AlunoMest (alunoID, médiaCurso)
- » AlunoDout (alunoID, temCAT)

Regras de Transformação

- **10ª Regra – Associação de Generalização (5/5)**
  - » Considerar todas as classes da hierarquia. Neste caso cada "tipo" de alunos tem identificação própria. Ex: L-1234 e M-1234



» Aluno (pessoaID, nome, morada, dtNasc, tipo) Discriminante

» AlunoLic (alunoLicID, notaAcesso, nrDisciplinas, ECTS, *pessoaID*)

Este tipo de Aluno não existe na realidade, pelo que não tem chave própria

» AlunoPG (pessoaID, cursoAnterior, títuloTese, *orientador*)

» AlunoMest (alunoMestID, médiaCurso, *pessoaID*)

» AlunoDout (alunoDoutID, temCAT, *pessoaID*)

Regras de *Mapping* para Relational (1/3)

Diagrama de Classes	Modelo Relacional
Classe A (com atributos simples)	Tabela A com chave primária aID
Associação binária entre as classes A e B, com uma associação L, e multiplicidade mA e mB	Tabela A com a chave primária aID e tabela B com a chave primária bID Se mA > 1 e mB > 1, a tabela L, que representa a associação, tem a chave primária (aID,bID) Se mA = 1 e mB > 1, a tabela B contém aID como chave estrangeira

Regras de *Mapping* para Relational (2/3)

Diagrama de Classes	Modelo Relacional
Associação binária entre as classes A e B, com uma associação L, e multiplicidade mA e mB (continuação)	Se mA = 1 e mB = 1, a tabela L da ligação pode ter como chave primária (aID) ou (bID). A tabela L pode ser absorvida pela tabela A ou B.
Agregações, composições	Podem ser tratadas da mesma forma que as associações binárias.

Regras de *Mapping* para Relational (3/3)

Diagrama de Classes	Modelo Relacional
Generalização (árvore de herança)	Existem 3 soluções possíveis: - Esmagamento das classes da hierarquia num único esquema relacional correspondente à superclasse - Considerar apenas esquemas correspondentes às subclasses - Considerar todas as classes da hierarquia
Associação N-ária entre N classes	N Tabelas com a chave primária IDn, onde *n* situado entre 1 e N A chave primária da tabela L que representa a associação corresponde ao conjunto (ID1, ID2, ID3, ...IDN). Se a multiplicidade de alguma classe for 1, o ID correspondente pode ser retirado do conjunto