



V - Regras de Transformação, ou *Mapping*, do Diagrama de Classes para o Nível Lógico

António Palma dos Reis

Aristides Sousa Mendes

Filipa Pires da Silva

Jesualdo Fernandes

Winnie Picoto



Índice

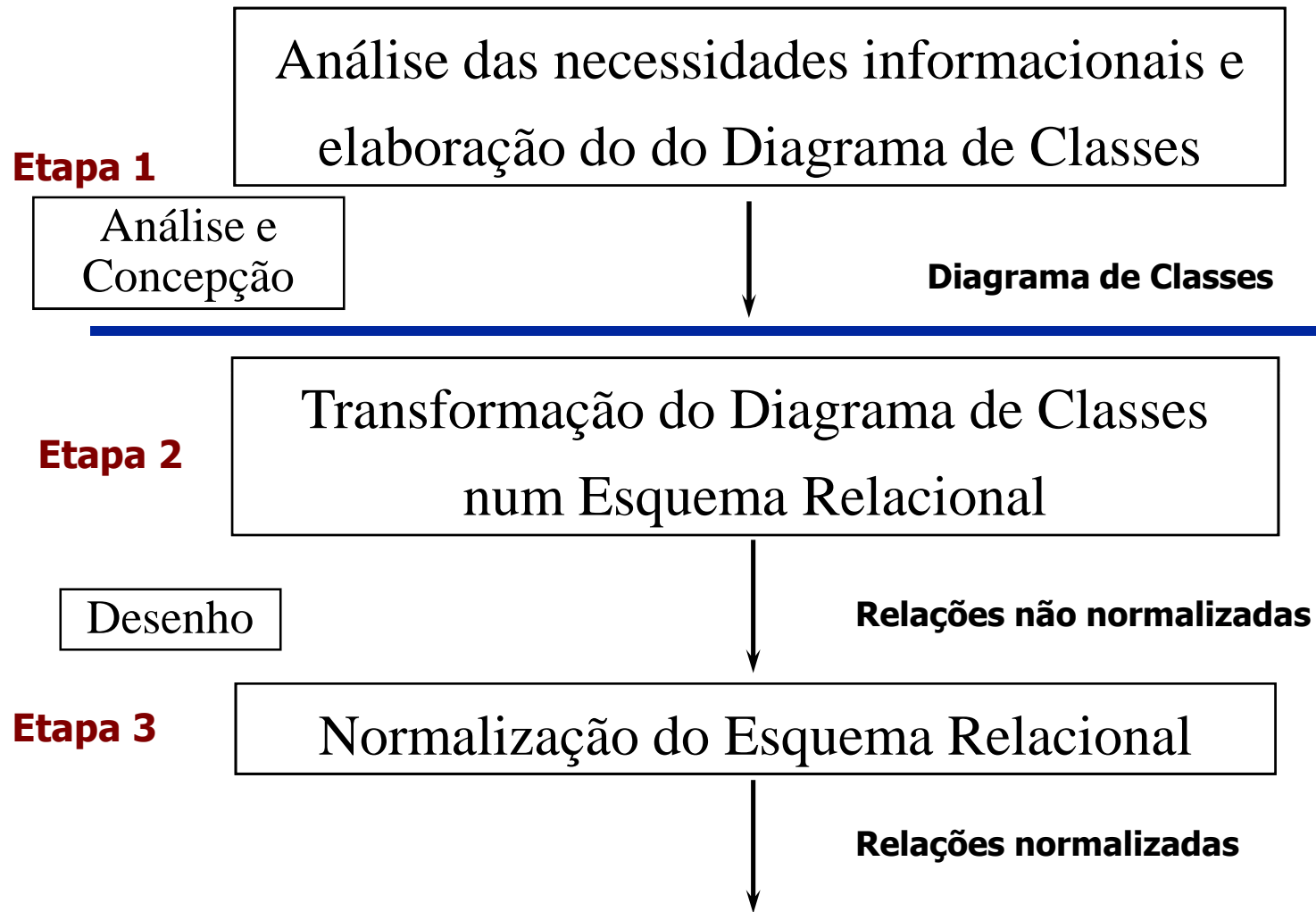
- Conceitos
- Regras de Transformação
- Normalização de Tabelas



- Depende da tecnologia de implementação:
BD relacionais ou BD OO
 - » Orientação para BD relacional: modelo relacional
 - » Orientação para BD OO: nível lógico OO (classes completas orientadas para BD OO e diagramas de sequência)



Etapas da Modelização da Informação





Conceito de “Relação” (1)

Relação – conjunto de atributos que respeitam determinados princípios e regras matemáticas.

Uma Relação pode ser representada em compreensão (através de *Esquemas de Relação*) ou em extensão (através de *Tabelas*).



Conceito de “Relação” (2)

- Ex de Esquema de Relação

EMPREGADO (número, nome, departamento)

- Ex de Tabela

Número	Nome	Depart
305	Pedro Ferraz	Contab
384	António Carvalho	Prod
392	João Silva	Prod
423	Carlos Pereira	Aprov
457	Fernando Azevedo	Prod
502	Elisa Teixeira	Contab
515	António Ferreira	Prod
533	Francisco Gomes	Prod



Conceito de “Regra de Transformação” para o Modelo Relacional

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



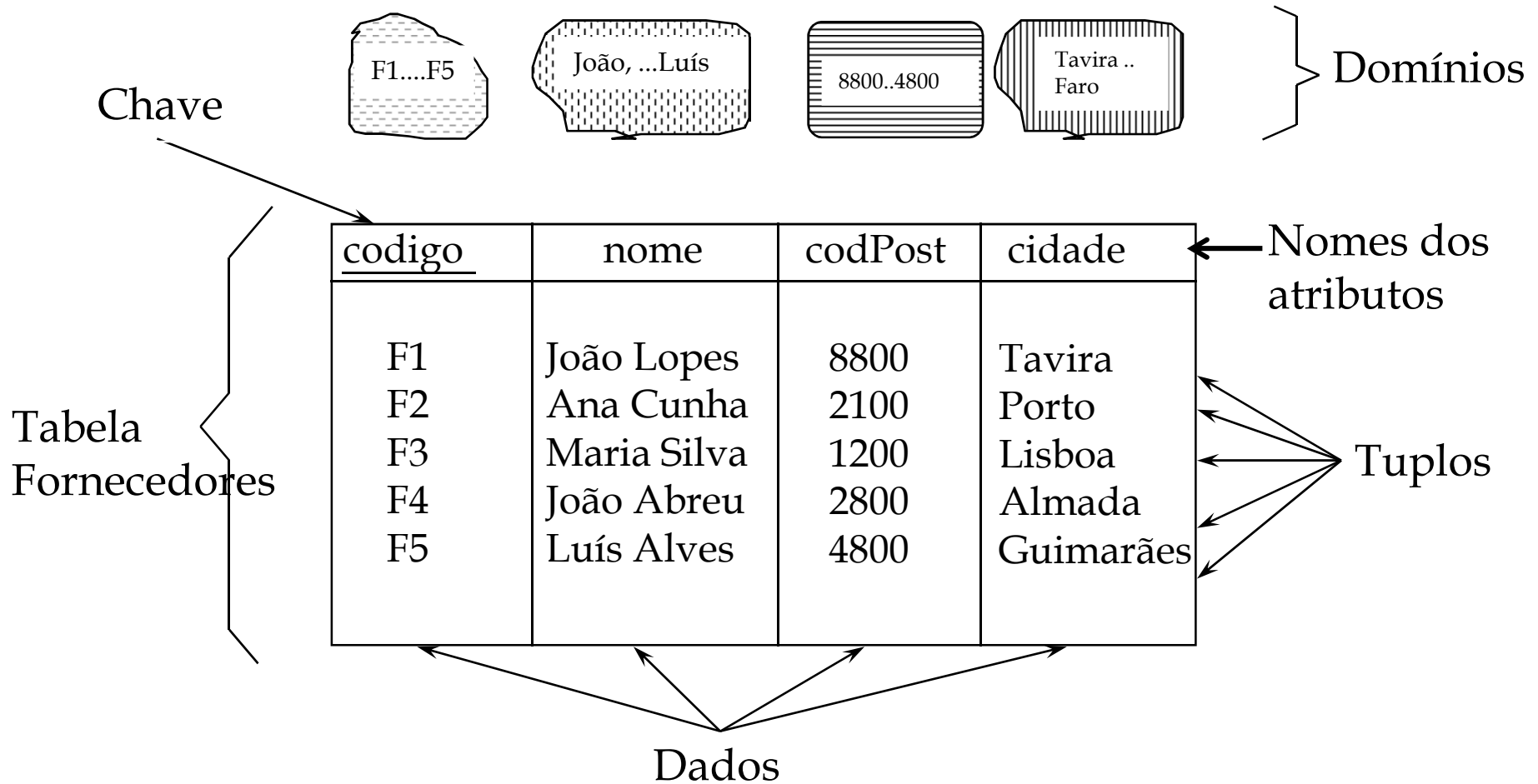
Conceitos do Modelo Relacional

Resumo dos conceitos da estrutura de dados do modelo relacional:

Termo Formal Relacional	Definição
Esquema de Relação	Relação representada em compreensão
Tabela	Relação representada em extensão
Tuplo	Linha ou Registo dum tabela (instância)
Chave Primária	Identificador Único (associação de um ou mais atributos)
Chave Estrangeira	Conjunto de um ou mais atributos que são chave numa outra tabela
Atributo	Nome de campo de uma relação ou de coluna dum tabela
Domínio de um atributo	Conjunto de valores possíveis do atributo



Exemplo dos conceitos:





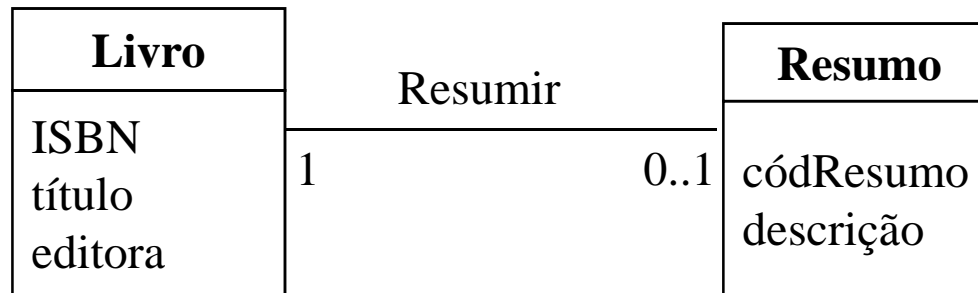
Chave Primária – todas as tabelas devem possuir uma chave primária

Valores Nulos – os atributos das relações não devem ter valores nulos (embora se aceitem valores em falta)

Redundância Controlada – alguns atributos aparecem repetidos, como chaves estrangeiras, mas tal repetição verifica-se apenas em atributos chave.



Associações binárias de 1..1 (1)



Atendendo a que a associação não é obrigatória de um dos lados, a transformação é feita para dois esquemas de relação:

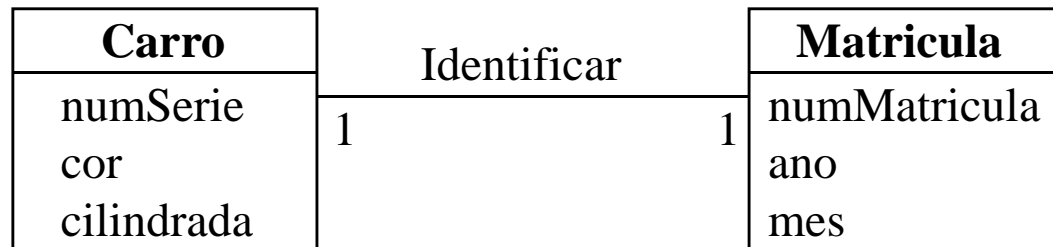
Livro (ISBN, título, editora)

Resumo (códResumo, descrição, *ISBN*).

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



Associações binárias de 1..1 (2)



Considerando que se trata de uma associação de 1:1, teríamos

Carro (numSerie, cor, cilindrada, *numMatricula*)

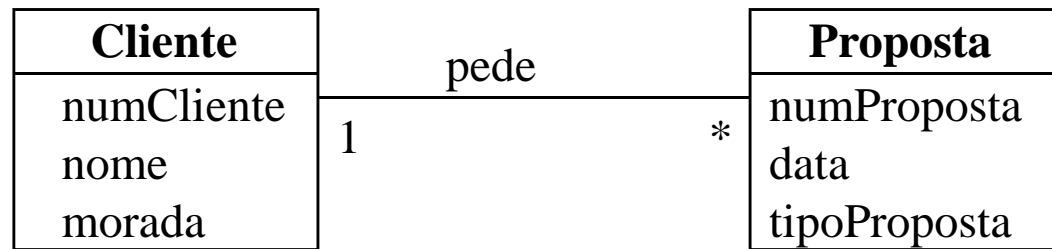
Matricula (numMatricula, ano, mes).

Como neste caso a associação é obrigatória em ambos os lados, os atributos devem ficar todos numa única relação:

Carro (numSerie, cor, cilindrada, numMatricula, ano, mes)



Associações binárias de 1..N



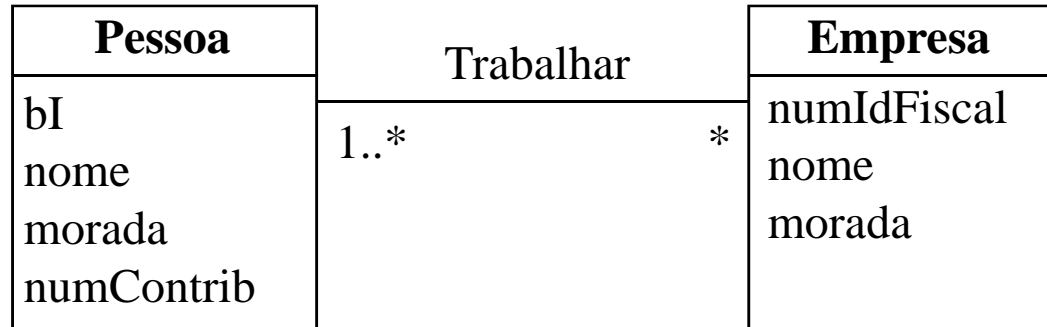
Tratando-se de uma associação de 1 para muitos, a relação do lado N recebe a chave estrangeira:

Cliente (numCliente, nome, morada)

Proposta (numProposta, *numCliente*, data, tipoProposta)



Associações de N:N



Tratando-se de uma associação de “muitos para muitos” – a transformação dá origem a uma 3^a tabela que representa a associação, cuja chave primária é composta pelas chaves das tabelas associadas:

Pessoa (bI, nome, morada, numContrib)

Empresa (numIdFiscal, nome, morada)

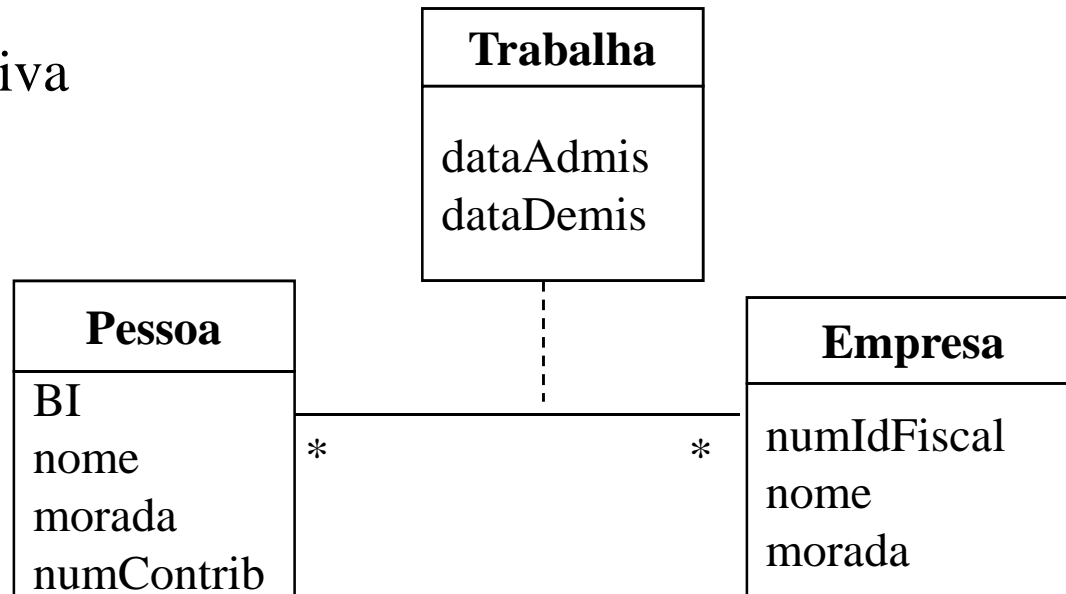
Trabalhar (*numIdFiscal*, *bI*)

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



Classes Associativas (1/2)

Exemplo: de classe associativa
numa associação de
“muitos para muitos”



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:

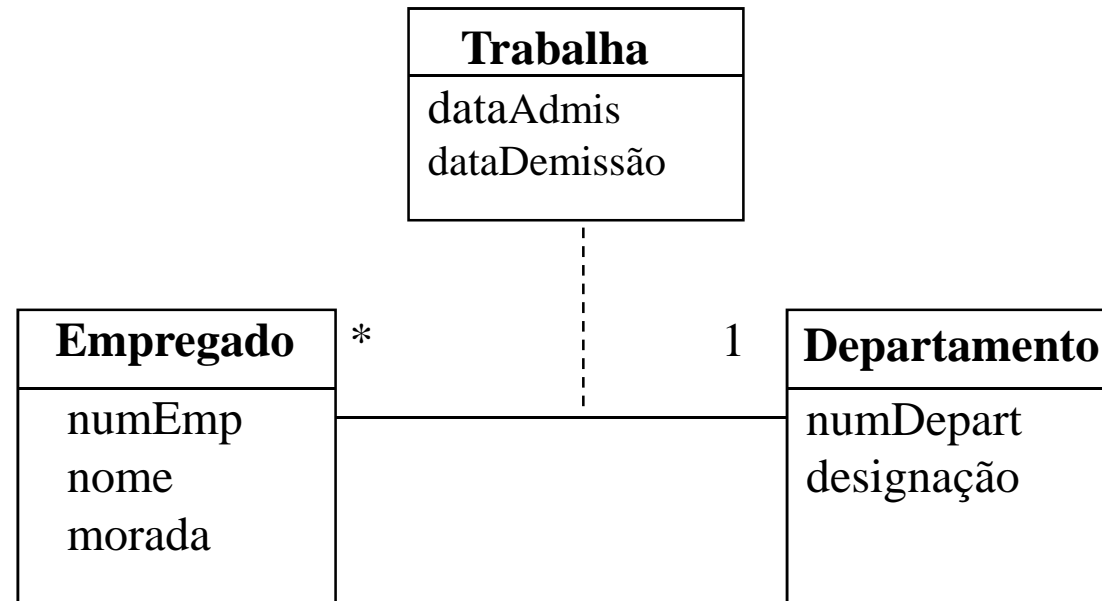
Pessoa (BI, nome, morada, numContrib)

Empresa (numIdFiscal, nome, morada)

Trabalha (numIdFiscal, BI, dataAdmis, dataDemis)



Classes Associativas (2/2)



Os atributos próprios da associação, tal como os atributos da classe do lado 1, ficam na relação referente à classe do lado N:

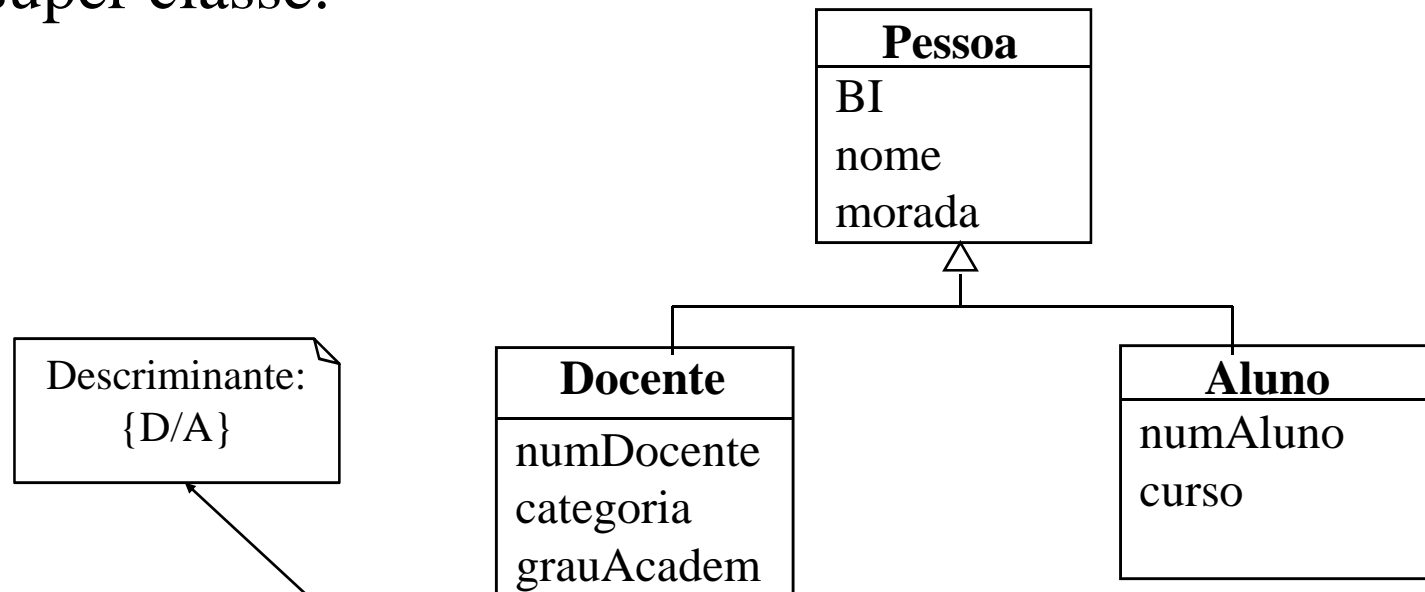
Departamento (numDepart, designação)

Empregado (numEmp, nome, morada, *numDepart*, dataAdmis, dataDemissão)



Generalizações

A transformação dá origem a uma tabela para cada uma das classes. O campo Tipo (discriminador) pertence à relação relativa à super classe.



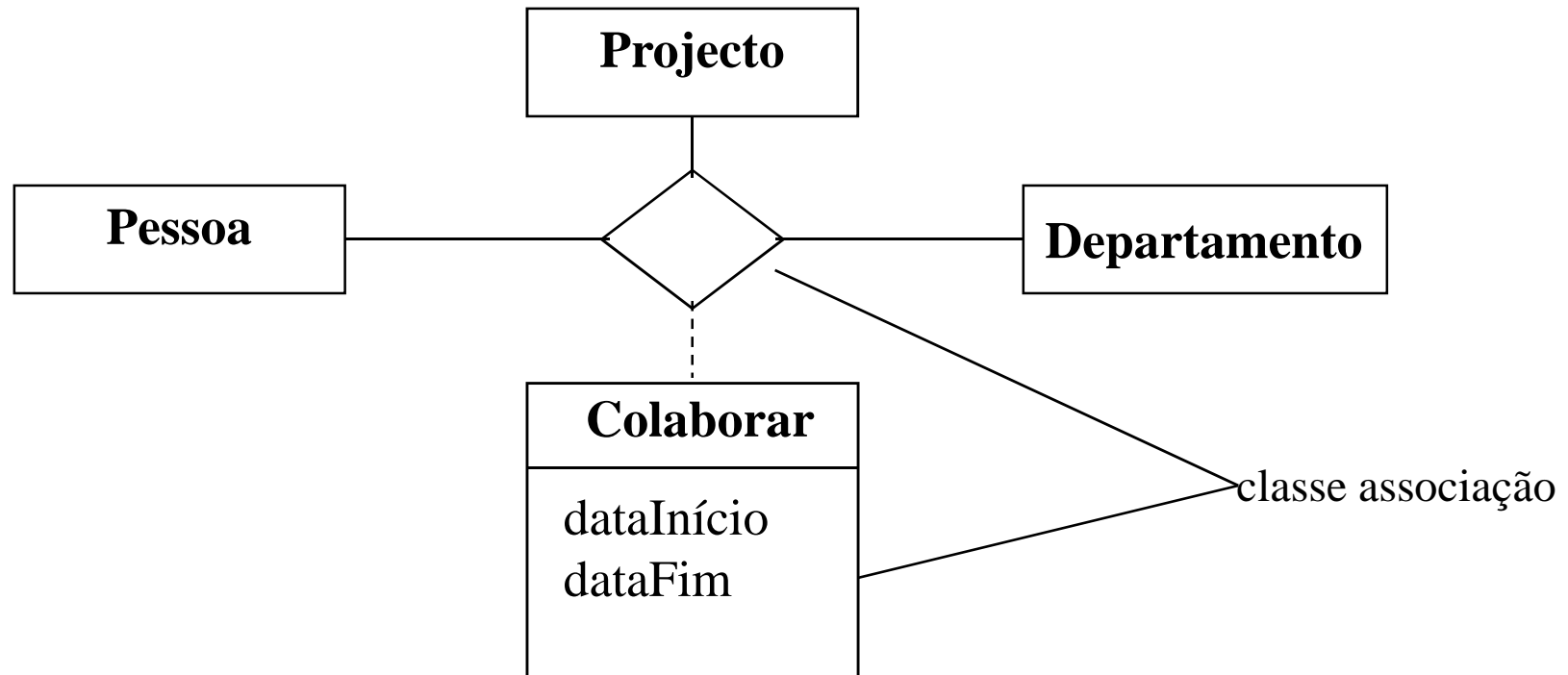
Pessoa (BI, nome, morada, D/A)

Docente (numDocente, categoria, grauAcadem, BI)

Aluno (numAluno, curso, BI)



Associações N-Árias ($N \geq 3$)



Pessoa (BI, nome, morada)

Departamento (numDepart, responsável, localização)

Projecto (numProjecto, responsável, dataInicio, dataFim)

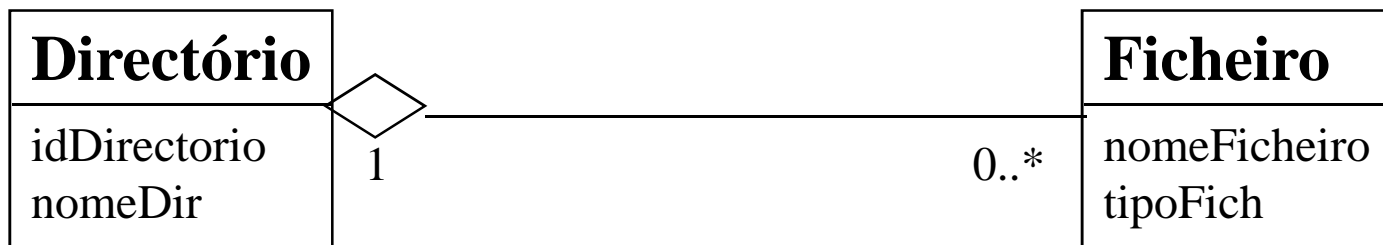
Colaborar (BI, numProjecto, numDepart, dataInicio, dataFim)



Agregações

Utiliza as mesmas regras da transformação para associações binárias com a mesma multiplicidade.

Exemplo:



Directório (idDirectório, nomeDir)

Ficheiro (nomeFicheiro, tipoFich, *idDirectório*)



Composições

A relação do lado da “parte” fica com a chave estrangeira integrada na sua chave primária.

Exemplo:



Livro (ISBNLivro, título)

Versão (ISBNLivro, númeroVersão, data)



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

Diagrama de Classes	Modelo Relacional
Classe A (com atributos simples)	Tabela A com chave primária A_ID
Associação binária entre as classes A e B, com uma classe associação L, e multiplicidade m_A e m_B	<p>Tabela A com a chave primária A_ID e tabela B com a chave primária B_ID</p> <p>Se $m_A > 1$ e $m_B > 1$, a tabela L da ligação tem a chave primária (A_ID,B_ID)</p> <p>Se $m_A = 1$ e $m_B > 1$, a tabela L da ligação se existir tem a chave primária (B_ID). A tabela L pode ser absorvida pela tabela B, contendo A_ID como chave estrangeira</p>



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

Diagrama de Classes	Modelo Relacional
Associação binária entre as classes A e B, com uma classe associação L, e multiplicidade m_A e m_B (continuação)	<p>Se $m_A > 1$ e $m_B = 1$, a tabela L da ligação tem a chave primária (A_ID). A tabela L pode ser absorvida pela tabela A, que contém B_ID como chave estrangeira</p> <p>Se $m_A = 1$ e $m_B = 1$, a tabela L da ligação pode ter como chave primária (A_ID) ou (B_ID). A tabela L pode ser absorvida pela tabela A ou B.</p>
Agregações, composições	Podem ser tratadas da mesma maneira que as associações binárias.



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

Diagrama de Classes	Modelo Relacional
Generalização (arvore de herança)	<p>Uma tabela para cada classe da arvore. Todas as tabelas na arvore usam a mesma chave primária. Os campos herdados das subclasses são armazenados na tabela da super classe.</p> <p>As tabelas da super classe em todos os níveis necessitam do campo adicional “tipo” para distinguir o tipo da sub classe para cada registo</p>
Associação N-ária entre N classes, com uma classe associação L	<p>N Tabelas com a chave primária ID_n, onde <i>n</i> situado entre 1 e N</p> <p>A chave primária da tabela L da ligação é feita do conjunto (ID₁, ID₂, ID₃, ...ID_N). Se a multiplicidade dos itens de alguma classe for 1, o ID correspondente pode ser retirado do conjunto</p>



Formas Normais

As relações devem respeitar as 3 seguintes Formas Normais:

- 1ª Forma Normal

- » Todos os seus atributos são atómicos (isto é, apresentam valores elementares e não conjuntos ou grupos repetitivos).

- 2ª Forma Normal

- » A relação está na 1FN e

- » todos os atributos não pertencentes à chave primária dependem da totalidade da chave, e não apenas de parte dela.

- 3ª Forma Normal

- » A relação está na 2FN e

- » Não há dependências transitivas, i.e., nenhum atributo é susceptível de ser definido por atributos não-chave, isto é, cada atributo deve depender apenas da chave primária da relação.



Aplicar as Regras de Mapping aos Diagramas de Classes dos acetatos a seguir



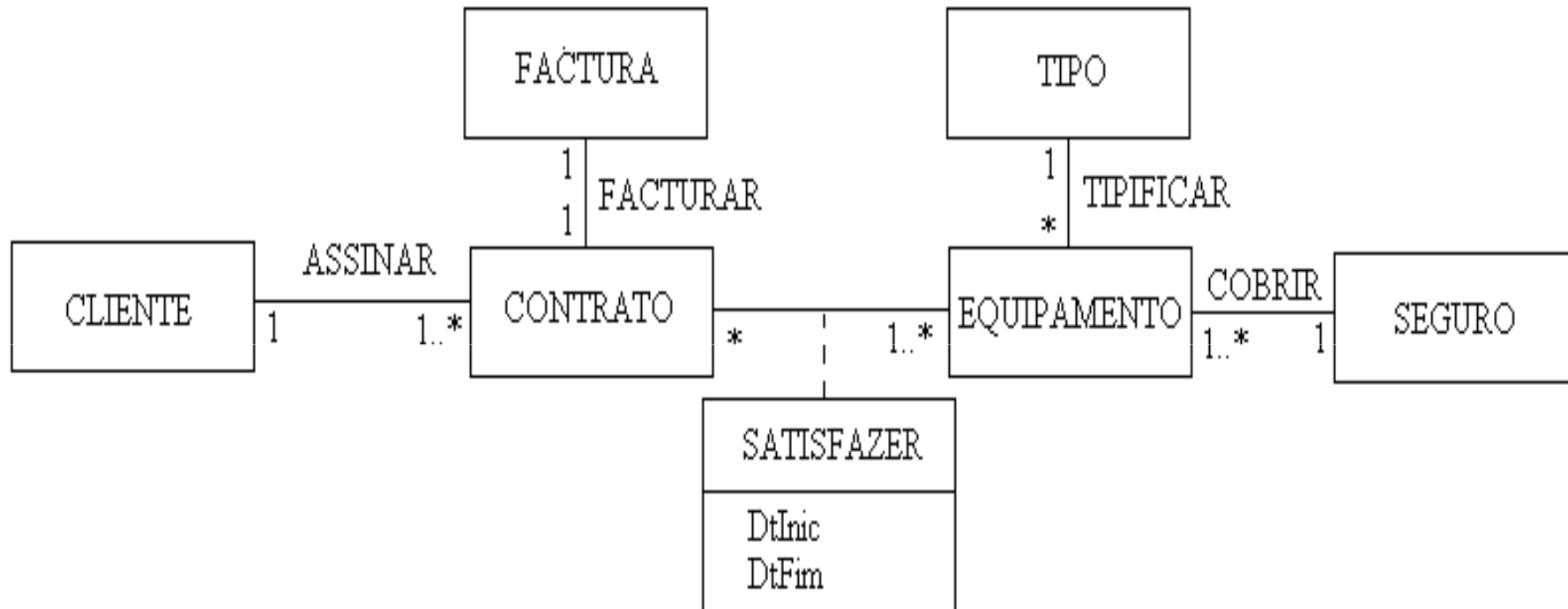
Caso de Estudo “Aluguer de Equipamento”

Pretende-se desenvolver um SI que permita gerir uma casa de aluguer de equipamento funcionando segundo os seguintes tópicos:

- o aluguer é formalizado por um contrato, de que constam, para além dos equipamentos a alugar e da data prevista de devolução, os elementos do seguro que será accionado em caso de qualquer avaria ou deterioração dos equipamentos alugados;
- se o cliente devolver os equipamentos fora do prazo definido, será sujeito a uma multa que depende do tipo de equipamento;
- no momento da devolução, o recepcionista verifica o estado dos equipamentos e o aluguer é facturado.



Caso de Estudo “Aluguer de Equipamento” – Diagrama de Classes





Caso de Estudo “Aluguer de Equipamento” – Modelo Relacional

CLIENTE (numCliente, nome, NIF)

EQUIPAMENTO (codigoEqp, designação, dtAquisição, valorDiário,
estado, *codTipo*, *NumApólice*)

CONTRATO (numCrt, data, dtPrevDevol, *numCliente*)

FACTURA (numFactura, data, *numCrt*)

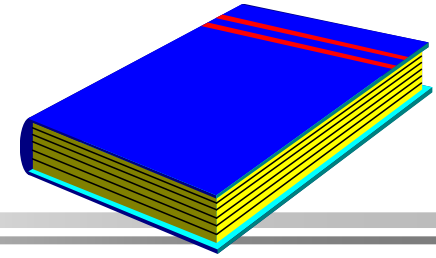
TIPO (codTipo, descrição, multaDiária)

SEGURO (apólice, dtInic, dtFim, franquia, custoSeguro)

SATISFAZER (*numCrt*, *codigoEqp*, dataInício, dataFim)



Bibliografia Utilizada



- 1 - Date, C. J.: “**An Introduction to Database Systems**”, Addison-Wesley, 1995.
- 2 - Oestereich, Bernd, *Developing Software With UML*, Addison Wesley, second edition, 2002.
- 4 - Palma dos Reis, A. e Sampaio, F., “*Transparências e Exercícios para a disciplina de Sistemas de Informação para a Gestão*”, Licenciatura de Gestão, ISEG, Lisboa, 2001.
- 5 - Sampaio, F.: *Sistemas de Gestão de Bases de Dados*, ISEG, Lisboa, 1996.
- 6 – Yun-Tung Lau, *The Art of Objects, Object-Oriented Design and Architecture*, Addison Wesley, 2001