



MAPEAMENTO OBJETO RELACIONAL

Professora Lucélia Oliveira

OS PROBLEMAS

- A Tecnologia orientada a objetos se consolidou como forma usual para desenvolver sistemas de software.
- A tecnologia de banco de dados relacionais é uma das tecnologias que tiveram maior êxito na área de computação.



DIFERENÇAS

- Os princípios básicos do paradigma da orientação a objetos e do modelo relacional são muito diferentes:
 - No modelo de objetos, os elementos (objetos) correspondem a abstrações que incluem dados e comportamentos.
 - No modelo relacional, os elementos correspondem apenas a dados.



CONCEITOS – MODELO RELACIONAL

- Relação: pode-se pensar em uma relação como uma tabela composta de linhas e colunas.
- Chave primária: é uma coluna ou conjunto de colunas cujos valores podem ser utilizados para identificar unicamente uma linha de uma relação
- Chave estrangeira: utilizadas para ligar duas relações. Deve existir, em uma das duas relações, uma coluna cujos valores fazem referência a valores de uma coluna da outra relação.



DIFERENÇAS

- O Modelo Relacional e o Diagrama de Classes não são equivalentes, as semelhanças são apenas semânticas:
 - O Modelo Relacional é um modelo de dados
 - O Modelo de Classes modela objetos que encapsulam dados e comportamentos



MAPEAMENTO DE CLASSES E ATRIBUTOS

- Classes são mapeadas para relações. Pode ser que várias classes sejam mapeadas para uma única relação ou que uma única classe seja mapeada para várias relações.
- Um atributo é mapeado para uma ou mais colunas.
- Nem todos os atributos são persistentes.
 - Exemplo: Pode ser que uma classe Pedido tenha um atributo derivado, total, utilizado para guardar um valor a ser pago por um pedido, mas que este atributo não seja armazenado no banco de dados.



ASSOCIAÇÕES

- Associações de Conectividade um para um
 1. A associação é obrigatória para ambos os extremos
 2. A associação é opcional para ambos os extremos
 3. A associação é obrigatória em um extremo e opcional no outro extremo
- Nos casos 1 e 2 a escolha da chave estrangeira é aleatória. No caso 3, deve-se optar pela relação que corresponde à classe de participação obrigatória.

Empregado(idEmpregado, matricula, CPF, nome, endereco, CEP)

Gerencia

Departamento(idDepartamento, sigla, nome, idEmpregadoGerente)



ASSOCIAÇÕES

- Associações de Conectividade um para muitos
- Neste caso deve adicionar uma chave estrangeira na relação de cardinalidade muitos referindo-se à chave primária da tabela de cardinalidade 1.

Empregado(idEmpregado, matricula, CPF, nome, endereço, CEP , idDepartamento)

Trabalha

Departamento(idDepartamento, sigla, nome, idEmpregadoGerente)



ASSOCIAÇÕES

- Associações Muitos-para-Muitos
- Neste caso uma nova relação deve ser criada.
- Há duas alternativas para se definir a chave primária da relação criada.
 - Uma chave primária composta pelas chaves primárias das duas relações participantes do relacionamento.
 - Criar uma coluna de implementação que sirva como chave primária simples da relação da associação.



AGREGAÇÕES E COMPOSIÇÕES

- Uma agregação é um tipo especial de associação.
- O mesmo procedimento para realizar o mapeamento de uma associação poderá ser usado para mapear uma agregação (ou composição).



AGREGAÇÕES E COMPOSIÇÕES

- A diferença semântica entre uma associação e uma agregação influi na forma como o SGBDR deve agir quando um registro da relação correspondente ao todo deve ser excluído ou atualizado.
- Isso pode ser implementado, por exemplo, utilizando-se recursos de um SGBDR, como *triggers* e *stored procedures*.
 - Exemplo: não faz sentido continuar mantendo os itens do pedido, de um pedido que já foi excluído.



CLASSES ASSOCIATIVAS

- Para cada classe associativa é criada uma nova relação associativa.



GENERALIZAÇÃO

- Existem três alternativas para se mapear relacionamentos de generalização:
 1. Uma relação para cada classe da hierarquia
 2. Uma relação para toda a hierarquia
 3. Uma relação para cada classe concreta da hierarquia



UMA RELAÇÃO PARA CADA CLASSE DA HIERARQUIA

- Neste caso, uma relação é criada para cada uma das classes
- Ponto forte
 - Esta é a alternativa que melhor reflete o modelo orientado a objetos.
- Ponto Fraco
 - Apresenta desvantagens com relação ao desempenho da manipulação das relações.



DESVANTAGENS

- Sempre que houver a inserção ou remoção de um objeto, seja ele de que subclasse for, deverá haver inserção ou remoção em duas tabelas, porque as colunas que representam objetos dessa classe estão divididos em duas relações.
- Algumas consultas precisam processar dados em que as colunas necessárias não estão na mesma relação.

Neste caso uma operação de junção (*join* – bastante cara do ponto de vista de desempenho) deverá ser usada.



UMA RELAÇÃO PARA TODA A HIERARQUIA

- Neste caso, cria-se uma única relação com colunas para representar os atributos da superclasse e também os atributos de todas as subclasses. Deve-se criar uma nova coluna para identificar a que classe da hierarquia pertence o objeto.
- Pontos fortes
 - É bastante simples e facilitam situações em que objetos mudam de classe.
- Pontos Fracos
 - Desperdício de espaço no armazenamento.



DESVANTAGENS

- Um novo atributo deve ser criado para a classe: toda a estrutura hierárquica sofre a modificação (mesmo sem necessidade)
- Desperdício de espaço de armazenamento: principalmente quando a classe possui várias irmãs. Os objetos pertencem a uma, e somente uma classe da hierarquia, mas, os atributos são comuns a todas as classes.



UMA RELAÇÃO PARA CADA CLASSE CONCRETA DA HIERARQUIA

- Neste caso, para cada subclasse é criada uma relação. Os atributos específicos de cada subclasse correspondem a coluna da relação correspondente e também possui colunas para cada atributo herdado da superclasse.
- Ponto forte
 - Agrupa os objetos de uma classe em uma única relação.
- Ponto Fraco
 - Ao modificar a superclasse, todas as relações sofrem modificações.

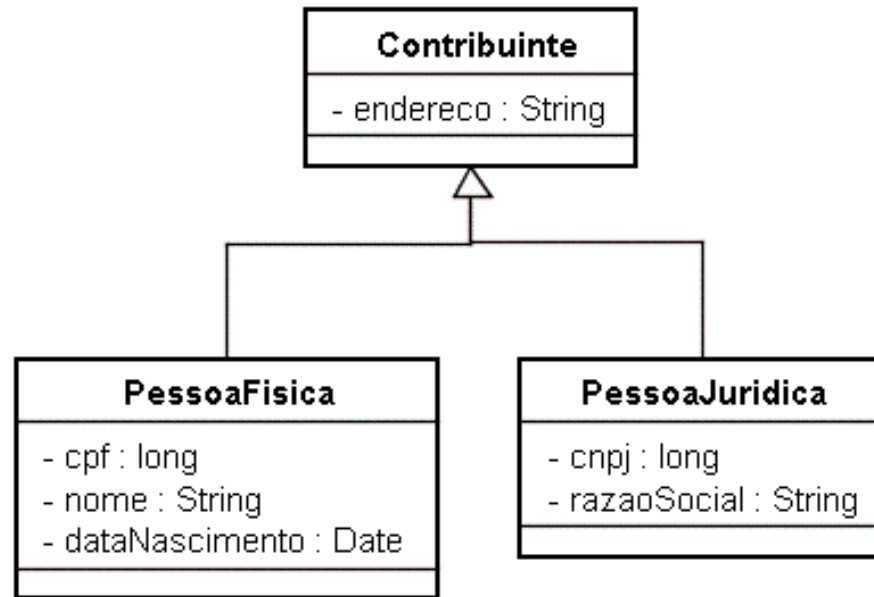


DESVANTAGEM

- Quando uma superclasse é modificada, cada uma das relações correspondentes às suas subclasses deve ser modificada.



EXEMPLO



1ª. Alternativa

Contribuinte(id, endereco)
PessoaFisica(id, nome, dataNascimento, cpf, idContribuinte)

2ª. Alternativa

Pessoa(id, nome, dataNascimento, cpf, cnpj, razaoSocial, tipo)

3ª. Alternativa

PessoaFisica(id, nome, dataNascimento, cpf, endereco)
PessoaJuridica(id, cnpj, razaoSocial, endereco)

CONCLUSÃO

- Nenhuma das alternativas de mapeamento de generalização é a melhor.
- Cada uma delas possui vantagens e desvantagens, e a escolha da alternativa a ser utilizada depende das características do sistema de software que está sendo desenvolvido.

