

Arquitura e Projeto de Sistemas

Mapeamento Objeto Relacional

Prof. Igor Avila Pereira
igor.pereira@riogrande.ifrs.edu.br

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Câmpus Rio Grande
Divisão de Computação

Agenda

- 1 Introdução
- 2 Problema
- 3 Solução
- 4 Conceitos
- 5 Técnicas de Mapeamento
 - Herança
- 6 Padrões de Mapeamento Objeto-Relacional
- 7 Considerações Finais
- 8 Links Interessantes

Introdução

Apresentaremos as técnicas de mapeamento objeto-relacional e soluções elegantes para o problema que aparece quando o sistema precisa armazenar de forma permanente os dados gerados por ele em um SGBD Relacional: **o descompasso de impedância**.

O ORM (ou *ORM*) nada mais é que o ato de conversão de objetos em memória para dados relacionais e vice-versa.

Introdução

Os SGBDR são considerados sistemas críticos na maioria das organizações, enquanto o modelo de objetos é adotado no desenvolvimento das aplicações de negócio.

Portanto, reforça-se a necessidade da utilização de metodologias que permitam que haja uma integração efetiva entre as interfaces, que tendem a ser orientadas à objetos e os dados, armazenados no modelo relacional.

Problema

Paradigma Relacional

De um lado...

Envolve a existência de tabelas uni-dimensionais de armazenamento de dados, onde cada linha ou tupla representa um determinado registro no banco de dados.

Problema

Paradigma O.O

De outro lado...

Objetos incluem estruturas de dados como listas e mapas que podem utilizar herança e ligações diretas entre outros objetos

- Além disso, os objetos são criados e modificados em memória, sendo necessário coordenar o que ocorre na memória com o que ocorre em disco .

Problema

Em síntese

De uma forma geral, na persistência de objetos em SGBD, a identidade dos objetos, seus relacionamentos (herança, agregação e associação) e seu estado devem ser preservados nas tabelas relacionais.

- É uma abordagem simples e eficiente de persistência, entretanto, retorna-se ao problema inicial da impedância entre a linguagem relacional e a linguagem O.O.

Solução

A criação de uma camada de persistência de objetos permite diminuir o acoplamento entre o banco de dados e a aplicação.

- Desta forma, uma mudança em um modelo pode ser traduzida no outro, sem que haja a necessidade de se reestruturar toda a aplicação

Através de ORM, é possível persistir os objetos em um SGBDR, obtendo-se todas as vantagens trazidas por ele, e evitando-se os problemas que aparecem ao realizar-se a persistência em O.O. utilizando um modelo relacional (impedância de modelos).

Conceitos

Para permitir a persistência de objetos em um SGBDR, algum acordo deve ser feito quanto à forma como os dados serão armazenados. O mapeamento deve envolver cada elemento de um objeto:

- seus atributos,
- relacionamentos e
- sua herança.

Logo, os conceitos da programação O.O. devem ser mapeados para estruturas de tabelas relacionais (agregação, associação, herança, polimorfismo).

Conceitos

Isso traz algumas questões, que devem ser consideradas no mapeamento entre os modelos e na escolha de um framework de ORM, entre elas

- Como converter os valores das colunas em objetos e apresentá-los no resultado das consultas?
- Como atualizar os dados caso o estado de um objeto mapeado seja alterado?
- Como modelar os relacionamentos entre os objetos?
- Como modelar a herança dos objetos no modelo relacional?
- Qual estratégia de caching pode ser utilizada para minimizar os acessos ao SGBD?
- Como executar funções agregadas?

Conceitos

A principal tarefa do mapeamento O/R envolve a identificação das construções da orientação a objetos que se deseja extrair do esquema relacional, entre elas a identificação das classes, dos relacionamentos que podem ser representados no modelo de objetos e estabelecer as cardinalidades.

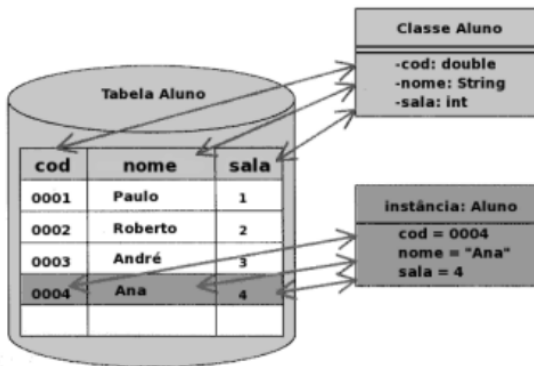
Como foi visto, o processo de mapeamento busca, basicamente, a tradução entre os modelos O.O. e relacional, partindo do princípio de que há uma arquitetura comum entre ambos.

Técnicas de Mapeamento

As principais técnicas de mapeamento de objetos em SGBDR podem ser descritas como:

- **Mapeamento classe-tabela:** Mapeamento de uma classe em uma ou mais tabelas, ou de uma tabela para uma ou mais classes, e mapeamento de herança;
- **Mapeamento atributo-coluna:** Mapeamento de tipos em atributos;
- **Mapeamento relacionamento - chave estrangeira:** Mapeamento dos relacionamentos O.O. em relacionamentos entre tabelas.

Técnicas de Mapeamento



Para modelos de dados muito simples, classes podem ser mapeadas diretamente em tabelas, em uma relação um-para-um.

Técnicas de Mapeamento

Esta é a forma mais intuitiva de mapeamento classe-tabela, onde todos os atributos da classe persistente são representados por todas as colunas de uma tabela no modelo relacional.

- Assim, cada instância do objeto pode ser armazenada em uma tupla da tabela.

Porém, este tipo de modelo pode conflitar com o modelo entidade-relacionamento existente, que pressupõe a normalização das tabelas e a otimização de consultas.

Técnicas de Mapeamento

Uma estratégia mais realista de mapeamento classe-tabela divide-as em duas categorias:

- **Mapeamento de subset**
- **Mapeamento de superset**

Técnicas de Mapeamento

- **Mapeamento de subset**, onde os atributos da classe persistente representam algumas ou todas colunas de uma tabela.
 - Esta estratégia convém para casos onde todos os atributos de uma classe persistente são mapeados a uma mesma tabela, e onde não há preocupação de incluir as colunas que não fazem parte do modelo de negócios.
 - Pode referir-se também à herança de tabelas simples.

Técnicas de Mapeamento

- **Mapeamento de superset**, onde os atributos da classe persistente são derivados de colunas de múltiplas tabelas.
 - Este tipo de mapeamento é usado para criar "classes de visão", que ocultam o modelo físico de dados, ou para mapear uma árvore de herança de classes utilizando o mapeamento vertical.

Técnicas de Mapeamento

Os atributos de uma classe, por sua vez, podem ser mapeados para zero ou mais colunas de uma tabela de um SGBDR, pois os atributos de um objeto não são necessariamente persistentes.

- Caso um atributo seja um objeto por si só, o mapeamento pode ser feito para várias colunas da tabela.
- Os atributos podem ser caracterizados em:
 - **Atributos Primitivos:**
 - **Atributos de Referência:**

Técnicas de Mapeamento

- **Atributos Primitivos:** Este termo denota um atributo de uma classe que é mapeado a uma coluna de uma tabela, isto é, refere-se ao valor de um tipo de dados específico (*int*, *float*, *double*, dentre outros).

Técnicas de Mapeamento

- **Atributos de Referência:** Atributos de referência representam relacionamentos com outras classes, isto é, referem-se a atributos cujo tipo é uma referência a outro objeto ou conjunto de objetos (composição).

Técnicas de Mapeamento

Em um modelo orientado a objetos, uma classe pode se relacionar com outra através de agregação ou associação.

- A cardinalidade (ou multiplicidade) de um relacionamento pode ser [1:1], [1:n], [n:1], [n:m].

Tabelas são relacionadas utilizando-se das mesmas cardinalidades.

Relacionamentos, no modelo de objetos, são implementados com combinações de referências a objetos e operações.

Técnicas de Mapeamento

Em O.O

Quando a multiplicidade é 1 (0..1 ou 1), o relacionamento é implementado por uma referência a um objeto ou operação de *get/set*.

Quando a multiplicidade é N (N , 0..*, 1..*), o relacionamento é implementado através de um atributo de coleção (por exemplo, um *array*), e por operações de manipulação deste *array*.

Técnicas de Mapeamento

Ainda em O.O....

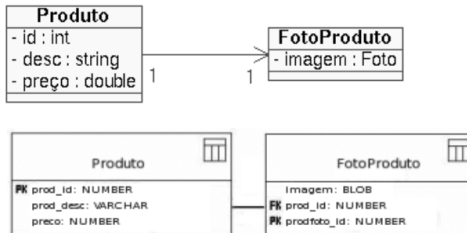
Sendo assim, as relações entre objetos são implementadas explicitamente através de atributos de referência, enquanto as relações entre tabelas são realizadas através de associações de chaves estrangeiras.

Um framework de ORM pode mapear as relações entre objetos utilizando as chaves estrangeiras das tabelas correspondentes.

Técnicas de Mapeamento

O mapeamento de associações preocupa-se, basicamente, com duas categorias de relacionamentos entre objetos. A primeira baseia-se na multiplicidade:

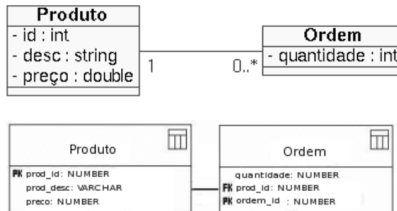
- **Mapeamento 1:1:** O relacionamento 1:1 é implementado através de restrições de chave estrangeira no modelo relacional e é sempre bi-direcional



Técnicas de Mapeamento

Mapeamento 1:n: O relacionamento 1:n pode ser implementado de forma similar ao mapeamento 1:1, onde a chave estrangeira é adicionada à classe associativa.

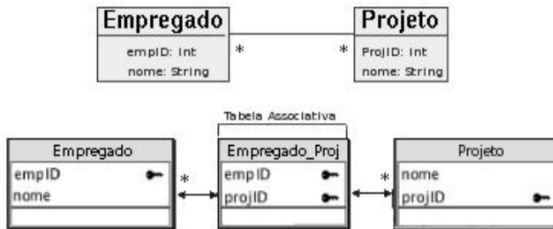
- Os atributos dos objetos agregados podem ser mapeados a uma única tabela, ou através da criação de uma tabela associativa para o tipo agregado.



Técnicas de Mapeamento

Mapeamento n:m: Para implementar relacionamentos n:m (muitos-para-muitos), que não existem fisicamente no modelo relacional, utiliza-se uma tabela associativa.

- O relacionamento passa a ser representado em uma tabela distinta no banco de dados, contendo os OIDs (ou chaves estrangeiras) dos objetos participantes da associação.



Técnicas de Mapeamento

A segunda categoria de relacionamentos baseia-se em dois tipos de "direcionalidade":

- relacionamentos unidirecionais e
- relacionamentos bidirecionais

O relacionamento unidirecional ocorre quando um objeto relaciona-se a outro, mas este segundo desconhece as classes associadas a ele.

O modelo relacional apenas trabalha com relacionamentos bidirecionais, inclusive, este é um fator de descompasso de impedância entre as tecnologias.

Técnicas de Mapeamento

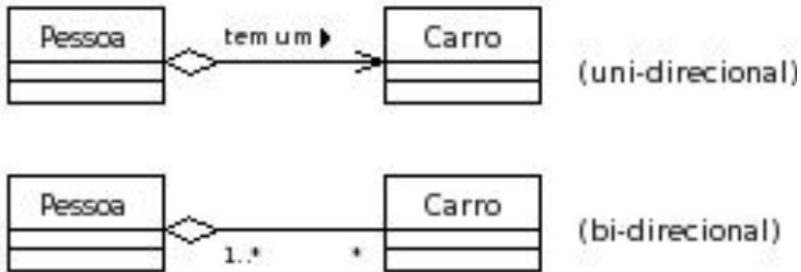


Figure: Direcionalidade

Técnicas de Mapeamento

Outra forma de relacionamento entre objetos são os relacionamentos recursivos

- **Por exemplo:** um time pode ser integrante de outros times.

O mapeamento em tabelas pode ser feito da mesma forma que um mapeamento $n:m$, isto é, através da criação de uma tabela associativa.

A diferença é que, neste caso, ambas colunas serão chaves estrangeiras da mesma tabela.

Técnicas de Mapeamento: Herança

Um aspecto essencial da tecnologia O.O. é a herança. A herança permite que dados e comportamentos de uma superclasse (classe base ou classe pai) sejam reaproveitados por subclasses (as classes derivadas da classe pai).

- Bancos de dados relacionais não possuem o conceito de herança: entidades não podem herdar atributos de outras entidades.

Técnicas de Mapeamento: Herança

Há diversas estratégias de mapeamento de uma estrutura de herança de um modelo de objetos para um modelo relacional, cada uma elaborada para lidar com um problema específico, entre elas:

- **Herança de tabela simples**
- **Herança horizontal**
- **Herança vertical**

Técnicas de Mapeamento: Herança

Herança de tabela simples (Single Table per Class Hierarchy): Na estratégia de herança de tabela simples (*flat*), uma tabela representa todas as classes da hierarquia.

- Essa é uma forma eficiente de mapear classes onde consultas são efetuadas no nível da classe base (superclasse).
- Entretanto, há uma perda de espaço considerável (valores nulos) e a necessidade de uma coluna (no caso, "type") para possibilitar a identificação das instâncias correspondentes a cada classe.

Técnicas de Mapeamento: Herança

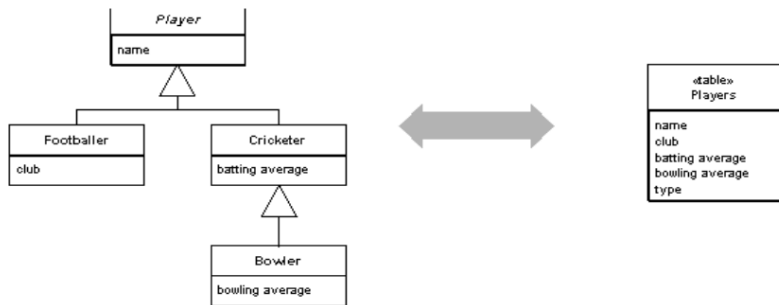


Figure: Herança Tabela Simples

Técnicas de Mapeamento: Herança

Herança horizontal (Table per Concrete Class): Na estratégia de herança horizontal, ou herança de tabelas concretas, cada classe concreta (não-abstrata) é associada à sua respectiva tabela, incluindo os atributos da classe herdada.

- A rapidez na persistência e alteração de instâncias é adquirida desta forma pelo custo da desnormalização.
- Qualquer alteração na classe abstrata deverá ser refletida nas tabelas das subclasses.

Técnicas de Mapeamento: Herança

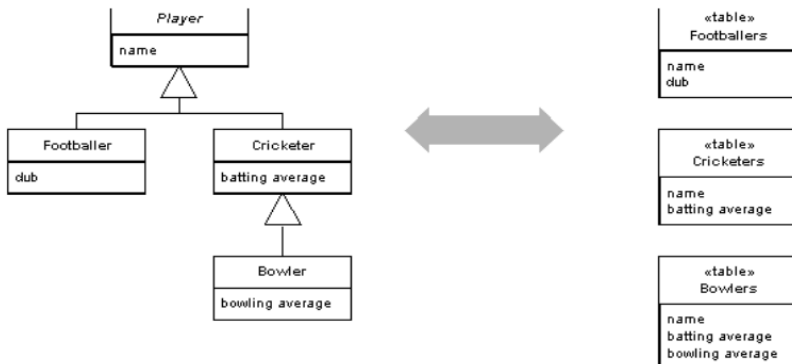


Figure: Herança Horizontal

Técnicas de Mapeamento: Herança

Herança vertical (Joined Subclass): Na estratégia de herança vertical, cada classe da hierarquia, inclusive classes abstratas, é associada a uma tabela separada no banco de dados.

- Através do mapeamento vertical, múltiplas tabelas são acessadas e todos os dados são extraídos para um objeto.
- Essa é a forma mais flexível de se lidar com dados legados complexos.

Porém, maior esforço será necessário para a reconstituição dos objetos, em comparação com as estratégias anteriores.

Técnicas de Mapeamento: Herança

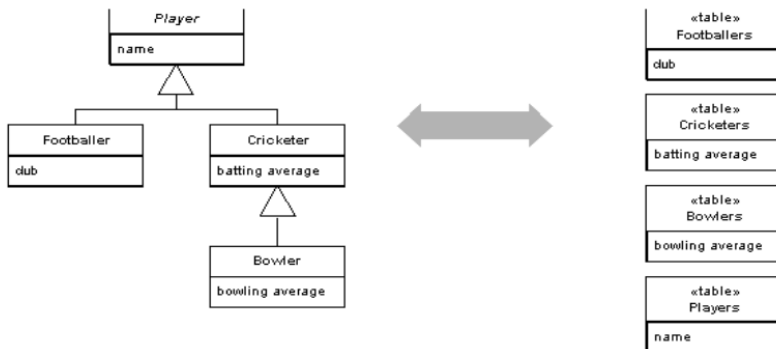


Figure: Herança Vertical

Padrões de Mapeamento Objeto-Relacional

Martin Fowler, autor de diversos livros a respeito de padrões de projeto, propôs alguns padrões específicos de ORM, utilizados na base de muitos frameworks, que serão descritos nesta seção.

Os padrões de ORM dividem-se entre padrões de arquitetura e padrões estruturais. A escolha básica para o padrão da arquitetura está dividida em quatro modelos:

- DAO
- Active Record
- E etc...

Padrões de Mapeamento Objeto-Relacional

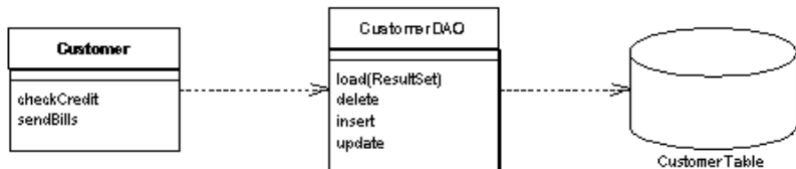


Figure: DAO

Padrões de Mapeamento Objeto-Relacional

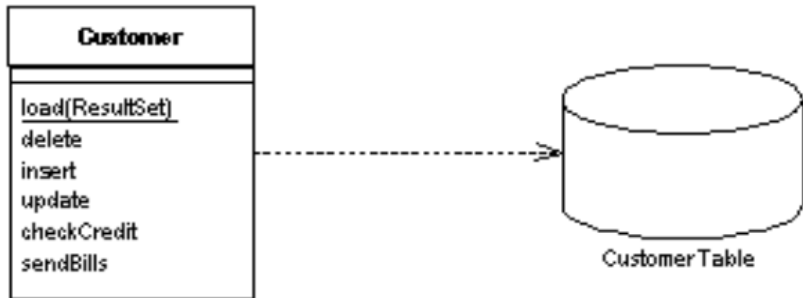


Figure: ActiveRecord

Considerações Finais

Através do mapeamento objeto-relacional, o código passa a ser muito mais consistente (eliminando a maioria dos problemas de impedância entre os modelos), simples de se manter e de ser estendido futuramente.

Um dos segredos do sucesso do mapeamento objeto-relacional é o entendimento de ambos paradigmas relacional e O.O.) e suas diferenças.

Links Interessantes

<http://martinfowler.com/eaCatalog/>

Arquitura e Projeto de Sistemas

Mapeamento Objeto Relacional

Prof. Igor Avila Pereira
igor.pereira@riogrande.ifrs.edu.br

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Câmpus Rio Grande
Divisão de Computação