

FRAMEWORK GERADOR DE TABELAS DE BANCO DE DADOS - CLONE WORKBENCH

Eduarda de Lima dos Anjos

delimadosanjoseduarda@gmail.com

Abstract. *This article presents the physical study of a database and a framework for the development of the physical model, with the main objective being the generation of tables of a database. For this article, we used: Workbench as Database, Fluid Interface as Design Pattern, Java as platform, Object Oriented Programming II and Database as base materials.*

Key-words: *Object Oriented Programming; Workbench; Fluent Interface; Java.*

Resumo. *O presente artigo realiza um estudo do modelo físico de um banco de dados e de um framework para o desenvolvimento do modelo físico, tendo como objetivo principal a geração de tabelas de um banco de dados. Para este projeto foram utilizados: o Workbench como Banco de Dados, Fluent Interface como Padrão de Projeto, Java como plataforma, Programação Orientada a Objetos II e Banco de Dados I como matérias base.*

Palavras-chave: *Programação Orientada a Objetos; Workbench; Fluent Interface; Java.*

1. Introdução

Atualmente a geração possui um nível crescente de informações, notícias e dados para serem armazenados da forma que acharem melhor, porém referenciando-se ao campo computacional, possuímos um método de persistência de dados. Na medida em que os tempos se passaram, a necessidade de armazenar informações de forma confiável e de fácil acesso aumentou, uma das soluções apresentadas para este tipo de armazenamento é o Banco de Dados onde se torna possível todo e qualquer armazenamento de informações de forma ordenada, de fácil acesso, tendo como princípio, integridade, confiabilidade, facilidade de manutenção e sobretudo, desempenho.

Utilizando um Banco de Dados para armazenar as informações, torna-se possível desenvolvê-lo com inúmeras finalidades, com uma devida sequência para criá-lo, realizando uma pesquisa referente aos requisitos, em seguida passando para o modelo conceitual (MER), em seguida o modelo Relacional e por fim, o modelo Físico.

O principal objetivo deste artigo é apresentar o SGBD utilizado, apresentar o desenvolvimento do Framework desenvolvido, o presente artigo irá abordar as matérias de Programação Orientada Objetos II e Banco de dados I do curso de Bacharelado de Ciência da Computação, Instituto Federal Catarinense, Campus Rio do Sul. Para o artigo presente, será utilizado o Banco de Dados MySQL, Modelo Físico, Padrão de Projeto Fluent Interface, plataforma Java,

Este artigo foi estruturado da seguinte forma: a Seção 2 irá apresentar as formas de persistências de dados abordadas nesta pesquisa, a Seção 3 diz respeito ao referencial teórico, Seção 4 irá apresentar uma classe referente ao clone do Workbench, Seção 5 será apresentada a conclusão.

2. Persistência de Dados

2.1 Banco de Dados

Segundo Korth, um banco de dados “é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico”, ou seja, sempre que for possível agrupar informações que se relacionam e tratam de um mesmo assunto, posso dizer que tenho um banco de dados.

2.1.1 Modelo Físico

O Modelo Físico descreve, por meio de alguma linguagem, como será feita a armazenagem no banco. Nesse nível se escolhe qual Sistema Gerenciador de Banco de Dados (SGBD) será usado, levando em consideração o modelo lógico adotado. Se o modelo Lógico é Relacional, podem ser utilizados SGBDs como por exemplo PostgreSQL ou MySQL.

2.2 Framework

Segundo Fayad e Schimidt (1997), um Framework é um conjunto de classes que colaboram para realizar uma responsabilidade para um domínio de um subsistema da aplicação. Um framework consiste em utilizar várias funcionalidades de um software em um conjunto de aplicações com problemas pertencentes a um mesmo domínio. Diferentemente das Bibliotecas, os Frameworks possuem como característica controlar o fluxo da aplicação.

2.3 Fluent Interface

De acordo com Pereira (2014): É um padrão de desenvolvimento de software orientado a objetos, que faz parte da categoria de DSL's (uma linguagem de domínio específico) internas. Melhora a legibilidade dos códigos, tornando o fluxo de de execução mais simples e intuitivo, torna o código menos extenso, fornece maior flexibilidade ao código. Esse modelo de desenvolvimento baseia-se no encadeamento de métodos, este encadeamento é possível através da transferência de contexto da instrução, fazemos essa transferência de contexto retornando uma referência ao objeto que realizou a primeira instrução.

3. Referencial Teórico

Segundo Ramez (2010 p.3,4) “[...]Um sistema gerenciador de banco de dados (SGBD) é uma coleção de programa que permite aos usuários criar e manter um banco de dados. O SGBD é um sistema de software de uso geral que facilita o processo de definição, construção, manipulação e compartilhamento de banco de dados entre diversos usuários e aplicações.”

4. Clone Workbench

A ideia proposta para realizar a criação de um framework é realizar um Clone do Workbench, de modo em que basta o usuário realizar a chamada de métodos referente ao database, linha e coluna, para salvar os parâmetros passados aos métodos poderá chamar o método para gerar o .sql. Este Clone possui um package chamado “Uteis” aonde possui classe referente ao select, count e group by.

Após desenvolvida a proposta e realizada uma pesquisa para verificar qual padrão de projeto seria aplicado no presente trabalho, será utilizado o padrão de projeto *Fluent Interface* pela agilidade e facilidade de uso que o padrão oferece.

No desenvolvimento deste projeto, foi realizado o desenvolvimento de nova classes, cinco referente ao DTO, uma referente a Interface e três referente a consulta e operação que serão permitidas a realização. Todas as classes que geram informações, utilizam a interface com o método gerador(), com suas individuais responsabilidades.

Como exemplo, iremos utilizar a classe CriaTabela:

1 2 3 4	<pre>public class CriaTabela implements IGerador{ private String nomeTabela; private List<CriaLinha> linhas = new ArrayList<>()</pre>
------------------	--

Anteriormente estão sendo apresentados os atributos da classe CriaTabela, como podemos verificar (Linha 1), a classe está realizando uma implementação da interface IGerador que possui o método gerador().

A seguir será demonstrado o método que realiza a validação e interage com o padrão *Fluent Interface*.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25	<pre>public String gerador(){ StringBuilder tabela = new StringBuilder(); tabela.append("CREATE TABLE IF NOT EXISTS"); tabela.append(" "); tabela.append(nomeTabela); tabela.append(" "); tabela.append("("); for (int i = 0; i < linhas.size(); i++) { tabela.append("\n"); if(i==linhas.size()-1){ tabela.append(linhas.get(i).gerador()); tabela.append("\n"); }else{ tabela.append(linhas.get(i).gerador()); tabela.append(","); } } tabela.append(")"); tabela.append(";"); return tabela.toString(); }</pre>
---	---

Importante ressaltar a nomenclatura que ele aplicará com base no atributo passado (linha 4), será utilizada a interface para gerar as linhas (linha 13), caso a primeira condição não seja atendida, será gerada a linha com base na interface (linha 16).

5. Conclusão

A seguir será apresentada a saída esperada após a execução do Clone Workbench.

```
CREATE DATABASE IF NOT EXISTS Loja DEFAULT CHARACTER SET UTF8;
USE Loja;

CREATE TABLE IF NOT EXISTS Funcionario (
  idFuncionario INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  nomeFuncionario VARCHAR(45) NOT NULL
);

CREATE TABLE IF NOT EXISTS Cargo (
  idCargo INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  NomeCargo VARCHAR(45) NOT NULL
);
```

Figura 1 – Saída esperada Workbench.

Fonte: Acervo do Autor.

O resultado obtido após o desenvolvimento do projeto foi gratificante, tendo atingido os objetivos principais do projeto que era clonar o workbench com seus comandos de criação de banco de dados, tabela, linhas e realizar validações. Foram implementados métodos para gerar o script com a extensão “.sql” o que irá facilitar na importação diretamente com o Workbench, métodos de consulta e operações. O framework fica aberto a extensão, sendo possível criar uma conexão com o banco de dados para estender automaticamente do framework e utilizar as operações JOINS.

6. Referências

ELMASRI, Ramez. **Sistema de Banco de Dados: Introdução**. 6. ed. São Paulo: Pearson Education do Brasil, 2010. 788 p. Tradução: Daniel Vieira.

FAYAD, Mohamed; SCHIMIDT, Douglas C. **Object-oriented application frameworks**, New York, ACM, 1997 38p.

PEREIRA, Francoes da Silva. **Interface Fluente**. 2014. Disponível em: <<https://pt.slideshare.net/FrancoesdaSilvaPerei/interface-fluente>>.

REZENDE, Ricardo. Conceitos Fundamentais de Banco de Dados: O que é Banco de Dados?. 2006. Disponível em:
<<https://www.devmedia.com.br/conceitos-fundamentais-de-banco-de-dados/1649>>.