

# Banco de Dados – Atividade Hands-on em Laboratório - 06

## HSQLDB - Driver JDBC - Prof. Dr. Aparecido Freitas

### 1. Introdução

Os **Sistemas Gerenciadores de Banco de Dados** disponibilizam queries através do driver **JDBC**, as quais são processadas em quatro passos:

- a) Interpretação (**parsing**) da consulta **SQL**;
- b) Compilação da consulta **SQL**;
- c) Otimização e geração do plano de consulta para busca dos dados;
- d) Execução da consulta otimizada, busca e retorno dos dados.

Assim, um **Statement** sempre irá passar pelos quatro passos acima para cada consulta SQL enviada ao Servidor de Banco de Dados.

O driver **JDBC** também fornece a classe **PreparedStatement**, a qual pré-executa os passos de (a) a (c). Com isso, ao se criar um **PreparedStatement** alguma pré-otimização é feita de imediato. O efeito disso é que, se você pretende executar a mesma consulta repetidas vezes, mudando-se apenas os parâmetros de cada uma, a execução com o emprego de **PreparedStatement** será **mais eficiente** e com **menos carga** sobre o Sistema Gerenciador de Banco de Dados.

Outra vantagem de se empregar **PreparedStatement** é que, se utilizados de forma adequada, ajudam a evitar ataques de **SQL Injection**.

No parâmetro que contém a query que deve ser passada para o **SGBD**, deve-se colocar **?** (pontos de interrogação) no lugar dos valores que queremos preencher. Por exemplo:

```
String sql = "INSERT INTO PRODUTOS (NOME, DESCRICAO) VALUES (?,?)";
```

Para o preenchimento das posições faltantes, correspondentes aos pontos de interrogação, usa-se os valores **1**, **2**, e assim por diante, para marcar os parâmetros de forma sequencial. Isso é feito pelo código abaixo:

```
stmt.setString(1,NOME);
```

```
stmt.setString(2,DESCRICAO);
```

Neste exercício, iremos repetir o código que faz inserção de dados no banco de dados, com o emprego de **PreparedStatement**.

```

package br.maua;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class TestaInsert {

    public static void main(String[] args) throws SQLException {

        try {

            String nome = "Nikon D850";

            String descricao = "Camera DSLR - Full Frame";

            Connection connection =
DriverManager.getConnection("jdbc:hsqldb:hsqldb://localhost:59999/db"
, "SA" , null);

            System.out.println("conexao ao HSQLDB feita com
SUCESSO ! ");

            String sql = "INSERT INTO PRODUTOS (NOME,DESCRICAO)
VALUES (?,?)";

            PreparedStatement stmt =
connection.prepareStatement(sql,Statement.RETURN_GENERATED_KEYS );

            stmt.setString(1,nome);

            stmt.setString(2,descricao);

            boolean resultado = stmt.execute();

            System.out.println("resultado = " + resultado);

            ResultSet resultSet = stmt.getGeneratedKeys();

            while (resultSet.next() ) {

                Integer id = resultSet.getInt("id");
                System.out.println("ID = " + id + " gerado
...");

            }

        }

    }
}

```

```

        resultSet.close();
        stmt.close();
    }

    catch (SQLException e) {
        System.out.println("Erro SQLException....");
    }

    catch ( Exception e) {
        System.out.println("Problemas na conexao ao
HSQLDB....");
    }
}
}

```