UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Introdução à conectividade Java com Banco de Dados usando JDBC

Programação Desktop

Prof. Fabrício M. Lopes
fabricio@utfpr.edu.br

Interface PreparedStatement

- A interface **PreparedStatement** permite o uso de uma instrução SQL pré-compilada.
- Então um objeto **PreparedStatement** pode ser usado para executar a instrução SQL pré-compilada de forma eficiente várias vezes, evitando **SQL injection**[1].

[1]https://wiki.postgresql.org/wiki/Sql_injection

Interface PreparedStatement - Exemplo

- *String query = "SELECT * FROM clientes WHERE idade > ?";*
- *PreparedStatement stat = conn.prepareStatement(query);*
- Cada símbolo **?** na instrução SQL denota uma **variável** que deve ser preenchida.
- Para isso, deve ser chamado um dos vários métodos definidos:
- *stat.setInt(1, idade);*
- *ResultSet res = pstdados.executeQuery();*

Interface PreparedStatement – Exemplo 2

```
String sqldml = "INSERT INTO clientes (id, nome, fone, email, idade) VALUES (?, ?, ?, ?, ?)";

pstdados = connection.prepareStatement(sqldml, ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE);

pstdados.setInt(1, 1001);

pstdados.setString(2, “João”);

pstdados.setString(3, “43-9999999”);

pstdados.setString(4, “joao@net.com”);

pstdados.setInt(5, 35);

int resposta = pstdados.executeUpdate();
```

Interface CallableStatement

- A interface **CallableStatement** é utilizada para a executar instruções SQL armazenadas no banco de dados na forma de **stored procedures ou functions**.
- Os parâmetros são referidos sequencialmente, por número, com o primeiro parâmetro sendo 1.

Interface CallableStatement - Exemplo

```
int tipo = ResultSet.TYPE_SCROLL_SENSITIVE;
int concorrencia = ResultSet.CONCUR_UPDATABLE;
CallableStatement cstdados =
connection.prepareCall(
"{call consultaclientesemail()}", tipo, concorrencia);
rsdados = cstdados.executeQuery();
```

Interface CallableStatement – Exemplo 2

```
int tipo = ResultSet.TYPE_SCROLL_SENSITIVE;
int concorrencia = ResultSet.CONCUR_UPDATABLE;

cstdados = connection.prepareCall(
                "{call consultamaioridade(?)}",
                tipo, concorrencia);

int idademinima = Integer.valueOf(
                JOptionPane.showInputDialog(
                "Entre com a idade minima: ", 18));

cstdados.setInt(1, idademinima);
rsdados = cstdados.executeQuery();
```

ResultSetMetaData

- Objeto que pode ser usado para obter informações sobre os tipos e propriedades das colunas em um objeto **ResultSet**.
- Quando é necessário acessar os dados sobre uma consulta realizada, como por exemplo ter um resultado definido a partir de uma tabela desconhecida, é possível saber os nomes das colunas, tipos de dados, etc.
- A classe **ResultSetMetaData** pode ser usada para descobrir as propriedades de um conjunto de resultados armazenados em um Resultset.
- Exemplo:
  - ResultSetMetaData metaData = result.getMetaData();

ResultSetMetaData

- Então é possível obter:
  - Número de colunas com o método getColumnCount(int i);
  - Nome da coluna com o método getColumnLabel(int i);
  - O tipo de dado armazenado pela coluna com o método getColumnClassName(int I);
  - Etc.
- Os índices para estes métodos começam com 1.

DatabaseMetaData

- Objeto que pode ser usado para obter informações sobre o banco de dados e suas propriedades em um objeto **Connection**.
- Exemplo:
  - DatabaseMetaData dbmt = conn.getMetaData();

DatabaseMetaData

- Então é possível obter:
  - Nome do BD: dbmt.getDatabaseProductName();
  - Versão do BD: dbmt.getDatabaseProductVersion();
  - URL do BD: dbmt.getURL();
  - Driver: dbmt.getDriverName();
  - Versão do Driver: dbmt.getDriverVersion();
  - Usuário: dbmt.getUserName();
  - Etc.

Classe java.util.Properties

- A classe **Properties** tem um método de leitura para ler um arquivo de pares chave/valor a partir de um fluxo. Exemplo:
  - Properties props = new Properties();
  - FileInputStream in = new FileInputStream(fileName);
  - props.load(in);
- O método ***getProperty()*** retorna o valor de uma determinada chave:
  - String driver = props.getProperty("jdbc.driver");

Programação com banco de dados em Java

- O gerenciamento de conexões pode ser implementado de forma simples e funcional por meio de uma classe utilitária.
- Exemplo:
  - JDBCUtil.java
  - configuracaobd.properties

Apresentação dos Exemplos

- AcessaBD2.java

Referências

- DEITEL, P.J. Java - Como Programar. Porto Alegre: Bookman, 2001.
- NIEMEYER, Patrick. Aprendendo java 2 SDK. Rio de Janeiro: Campus, 2000.
- MORGAN, Michael. Java 2 para Programadores Profissionais. Rio de Janeiro: Ciência Moderna, 2000.
- HORSTMANN, Cay, S. e CORNELL, Gary. Core Java 2. São Paulo: Makron Books, 2001 v.1. e v.2.