

JDBC

Avançado



JDBC

- Drivers
- Factory
- Prepared statements
- Meta-dados do conjunto de resultados
- Cursores e concorrência do result set
- Stored Procedures
- RowSet (Interface)
- Transações



Driver

- Até a versão 3
 - Necessidade de registrar o driver JDBC
 - Método `Class.forName`
- A partir do JDBC 4 (atualmente 4.2 no Java 8) esse procedimento não é mais necessário



Factory

- Considerando a possibilidade de conexões a diferentes bancos de dados, podemos implementar uma “fábrica” de conexões
 - Transparência para a aquisição dos programadores

```
public class ConnectionFactory {  
    public Connection getConnection() {  
        try {  
            return DriverManager.getConnection(  
                "jdbc:mysql://localhost/fj21", "root", "");  
        } catch (SQLException e) {  
            throw new RuntimeException(e);  
        }  
    }  
}
```



Prepared Statement

- SQL Injection

```
String sql = "insert into contatos " +  
            "(nome,email,endereco,dataNascimento) " +  
            "values (?, ?, ?, ?)";  
PreparedStatement stmt = connection.prepareStatement(sql);
```

- Passagem de parâmetros
 - stmt.setString(indice, valor);
- Execução do comando mais rápida, código mais limpo e seguro.



Meta-dados da Consulta

- Ao realizar uma consulta é possível obter informações sobre a sua execução
 - Número de colunas retornadas
 - Nomes dos campos das colunas retornadas
 - Tipo de dado esperado de cada coluna
- Aplicação:
 - Método `getMetaData()` de `ResultSet`
 - Retorna um objeto da classe `ResultSetMetaData`



Meta-dados da Consulta

```
ResultSetMetaData meta = rs.getMetaData();  
int colCount = meta.getColumnCount();  
  
// Imprime um cabeçalho para os resultados.  
for (int i = 0; i < colCount; i++) {  
    String colName = meta.getColumnName(i + 1);  
    System.out.print(' ');  
    System.out.print(colName);  
}  
System.out.println();
```



Cursos

- ResultSet oferece um cursor que permite a navegação pelos resultados
- Na especificação do Statement é possível especificar o tipo do cursor e o modo de concorrência

```
System.out.println("Creating statement...");  
stmt = conn.createStatement();
```



Cursors

- Tipos de cursores
 - TYPE_FORWARD_ONLY: só navega para frente
 - TYPE_SCROLL_INSENSITIVE: navega em qualquer direção
 - TYPE_SCROLL_SENSITIVE: navega em qualquer direção e atualiza os dados se houver alteração
- Tipos de concorrências
 - CONCUR_READ_ONLY: não pode ser atualizado
 - CONCUR_UPDATABLE: alterações replicadas no banco de dados



Cursores

- Exemplo: TYPE_SCROLL, métodos
 - next()
 - previous()
 - absolute(linha)
 - first()
 - last()
 - getRow()



Stored Procedures

- Procedimentos armazenados no banco de dados e executados remotamente
 - Para a execução deve ser invocado o método `prepareCall(nome da stored procedure)` que retorna um statement a ser executado (classe `CallableStatement` que estende `PreparedStatement`)



RowSet

- Interface criada a partir da versão 5 para facilitar as etapas de conexão e consulta aos BDs
- Dois tipos principais:
 - JdbcRowSet: result set atualizável
 - CachedRowSet: result set desconectado, lê os dados, desconecta e armazena em cache
- Em cache pode ser serializado, mas tem limite de memória



RowSet

```
JdbcRowSet rowSet = new JdbcRowSetImpl();  
rowSet.setUrl(DB_URL);  
rowSet.setUsername(USER);  
rowSet.setPassword(PASS);  
rowSet.setCommand(sql);  
rowSet.execute();  
//a pelos resultados.  
while (rowSet.next()) {  
    //ao ResultSet... */  
    String id = rowSet.getString("medico");  
    String area = rowSet.getString("area");  
    String first = rowSet.getString("nome");  
  
    System.out.print("ID: " + id);  
    System.out.print(", Area: " + area);  
    System.out.print(", Nome: " + first + "\n");  
}
```



Transações

- JDBC funciona com commit automático de transações
- Para desabilitar, use o método `setAutoCommit(false)` no objeto `Connection`
- Para confirmar use `commit()` em `Connection`
- Para cancelar, use `rollback()` em `Connection`
- Importante: o SGBD e seu driver devem oferecer suporte a essa funcionalidade

