



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DO RIO GRANDE DO NORTE
CAMPI JOÃO CÂMARA

PROGRAMAÇÃO COM ACESSO A BANCO DE DADOS CONEXÃO E TRANSAÇÃO

Nickerson Fonseca Ferreira
nickerson.ferreira@ifrn.edu.br

Conexão com Banco de dados

2

- Quais os problemas com a implementação atual ??
- Se esse sistema tiver 300 telas que acessam o BD ??
- E se eu mudar o meu SGBD ??
 - Teremos que alterar o código de todos os acessos
- Qual a solução para esse problema ??
 - CENTRALIZAR A CONEXÃO COM O BD NUMA ÚNICA CLASSE

Classe Conexao.java

3

```
public class Conexao {  
  
    public static Connection getConnection() {  
  
        Connection conexao = null;  
        try {  
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");  
            conexao = DriverManager.getConnection(  
                "jdbc:sqlserver://localhost:1433;database=teste",  
                "sa", "123456");  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
        return conexao;  
    }  
}
```

Instância da classe Connection

4

...

```
Connection conn = Conexao.getConnection();
```

```
Statement stmt = conn.createStatement();
```

...

Mais sobre conexão

5

- ❑ E agora ?? Está tudo correto com a aplicação de vocês ??
- ❑ Vocês fecham todas as conexões que abrem ??
- ❑ Um SGBD suporta uma determinada quantidade de conexões abertas simultaneamente.
- ❑ Quando essa quantidade é atingida e tentamos criar uma nova conexão, um erro é lançado.
- ❑ Para corrigir esse erro, basta fechar TODAS as Connections, Statements e ResultSet.

Fechando as conexões

6

- `resultado.close();`
 - Usado para fechar um ResultSet.
- `stmt.close();`
 - Usado para fechar um Statement.
- `conn.close();`
 - Usado para fechar uma conexão.
- Sempre devemos fechar nessa ordem, o último elemento fechado deve ser a conexão.

Fechando as conexões

7

```
Connection conexao = null;
Statement stmt = null;
try {
    conexao = Conexao.getConnection();
    stmt = conexao.createStatement();
    int retorno = stmt.executeUpdate(comandoInsert);
} catch (SQLException ex) {
    JOptionPane.showMessageDialog(null,
        "Ocorreu um erro ao acessar o banco de dados!", null,
        JOptionPane.ERROR_MESSAGE);
} finally {
    try {
        if (stmt != null) stmt.close();
        if (conexao != null) conexao.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

Transações

8

- Coleção de operações que desempenha uma função lógica única no BD
- Imagine um sistema bancário onde é necessário realizar uma transferência de uma conta A para uma conta B
- Existem algumas características que uma transação deve ter.

Atomicidade

Consistência

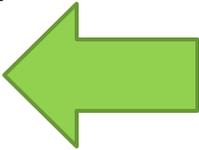
Isolamento

Durabilidade

ACID

Java e Transações

9

- O controle das transações pode ser gerenciado através da aplicação
- Podemos realizar essa gestão utilizando:
 - API específica: JTA
 - JDBC 

Java e Transações

10

- Existem alguns métodos responsáveis pelo gerenciamento de transações:
 - `conexao.setAutoCommit(boolean);`
 - `conexao.commit();`
 - `conexao.rollback();`

Java e Transações

11

```
Connection conexao = null;
Statement stmt = null;
try {
    conexao = Conexao.getConnection();
    conexao.setAutoCommit(false);
    stmt = conexao.createStatement();

    for (int i = 0; i < 10; i++){
        stmt.executeUpdate(comandoInsert);
    }

    conexao.commit();
} catch (SQLException ex) {
    try {
        conexao.rollback();
    } catch (SQLException ex1) {}
} finally {
    try {
        if (stmt != null) stmt.close();
        if (conexao != null) conexao.close();
    } catch (SQLException e){
        e.printStackTrace();
    }
}
```

EXERCÍCIO

12

- Criar a classe `Conexao.java`
- Criar uma tela que simule uma transação de transferência de uma quantia da conta A para conta B.
- Os passos para uma transferência são:
 - Verifica saldo da conta de débito. Se não tiver saldo lança um erro. Caso contrário, vá para o passo seguinte.
 - Retira a quantia da conta de débito.
 - Adiciona a quantia na conta de crédito.
 - Realiza o commit da transação.
 - Se acontecer um erro, realiza o rollback.