

BANCO DE DADOS COM JAVA (Foco: Banco de Dados Mysql)

REQUISITOS: CONNECTOR J

WAMP SERVER. (<http://www.wampserver.com/en/index.php>)

O driver Connector/J permite trabalhar desde Java com bases de dados MySQL.

Mysql Connector é um driver criado por Mysql AB que te permitirá trabalhar com Mysql desde programas escritos em Java. A diferença em relação a outros drivers, é que este é de livre distribuição, e tem um bom rendimento.

MySQL Connector/J é um driver nativo de Java que converte as chamadas geradas por JDBC no protocolo de rede que utiliza a base de dados Mysql. Permite trabalhar com a linguagem de programação Java e desta forma construir programas que interactuam com Mysql.

O MySQL Connector/J é um driver JDBC tipo IV e contem todas as características de JDBC para utilizar Mysql.

No desenvolvimento das últimas versões incrementou-se bastante a velocidade do driver, ganhando tanto em rapidez como em eficácia. O driver suporta resultados de dados "streaming" o que permite ao usuário recolher um grande número de filas sem a necessidade de utilizar um buffer de memória. O driver implementa um protocolo de packs grandes que permitem enviar filas e campos BLOBs de até 2 GigaBytes. Na versão de desenvolvimento do mesmo implementam-se as novas características acrescentadas no API 3.0 de JDBC.

Download do Connector J: <http://www.mysql.com/downloads/api-jdbc-stable.html>

1) CONSULTANDO UMA TABELA DO MYSQL (SELECT)

```
System.out.println("Testando acesso a banco de dado MySQL\n\n");
```

```
Connection conn = null;
```

```
String teste = "SELECT * FROM tabela";
```

```
try {
```

```
Class.forName("com.mysql.jdbc.Driver");
```

```
conn =
DriverManager.getConnection("jdbc:mysql://localhost/bd?user=root");

System.out.println("A conexão foi um sucesso\n");

Statement stm = conn.createStatement();

ResultSet rs = stm.executeQuery(teste);

while (rs.next()) {

String nome = rs.getString("campo1");

String codigo = rs.getString("campo2");

String nota1 = rs.getString("campo3");

}

System.out.println("\n\nConsulta realizada com sucesso!!!\n");

} catch(ClassNotFoundException e) {

System.out.println("excessão Classe não encontrada");

e.printStackTrace();

} catch(SQLException e) {

System.out.println("SQL Exception... Erro na consulta:");

e.printStackTrace();

} finally {

try {

conn.close();

System.out.println("\n\nFechamendo a conexão");

} catch(SQLException erro) {

System.out.println("Erro no fechamento");

erro.printStackTrace();

}

}
```

PASSO A PASSO PARA CONEXÃO COM BANCO DE DADOS

1. INDICAR QUEM É O DRIVER

```
Class.forName("com.mysql.jdbc.Driver");
```

2. CONECTAR AO BANCO DE DADOS

```
Connection conn =  
DriverManager.getConnection("jdbc:mysql://localhost/bd?user=root");
```

3. CRIAR A CONSULTA (STATEMENT)

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

4. EXECUTAR A CONSULTA (QUERY)

```
ResultSet rs = stm.executeQuery("CONSULTA");
```

5. MANIPULAR OS RESULTADOS (TABELAS)

```
while (rs.next()) {  
  
String nome = rs.getString("campo1");  
  
String codigo = rs.getString("campo2");  
  
String nota1 = rs.getString("campo3");  
  
}
```

5. FECHAR A CONEXÃO

```
conn.close();
```

PS. TODOS OS COMANDOS RELACIONADOS COM BANCO DE DADOS DEVEM ESTAR DENTRO DE BLOCOS TRY-CATCH.

```
try {  
  
//CODIGO  
  
} catch(ClassNotFoundException e) {  
  
System.out.println("excessão Classe não encontrada");  
  
} catch(SQLException e) {
```

```
System.out.println("SQL Exception... Erro na consulta:");

} finally {

try {

conn.close();

System.out.println("\n\nFechamendo a conexão");

} catch(SQLException erro) {

System.out.println("Erro no fechamento");

}

}
```