

JDBC

Prof. Márcio Bueno (poonoite@marciobueno.com)


JBDC

- ▶ **“Java Database Connectivity ou JDBC é um conjunto de classes e interfaces (API) escritas em Java que faz o envio de instruções SQL para qualquer banco de dados relacional; Api de baixo nível e base para api’s de alto nível; Amplia o que você pode fazer com Java; Possibilita o uso de bancos de dados já instalados; Para cada banco de dados há um driver JDBC que pode cair em quatro categorias.”** <http://pt.wikipedia.org/wiki/JDBC>
- ▶ **Ponte JDBC-ODBC**
- ▶ **Driver API-Nativo**
- ▶ **Driver de Protocolo de Rede**
- ▶ **Driver nativo**

Hsqldb

- ▶ Banco de dados relacional OpenSource totalmente escrito em Java
- ▶ Muito pequeno e não precisa instalá-lo na máquina, basta ter o arquivo “hsqldb.jar” junto com o seu projeto
- ▶ Pode rodar em modo “isolado” ou “servidor”

Hsqldb - Configuração

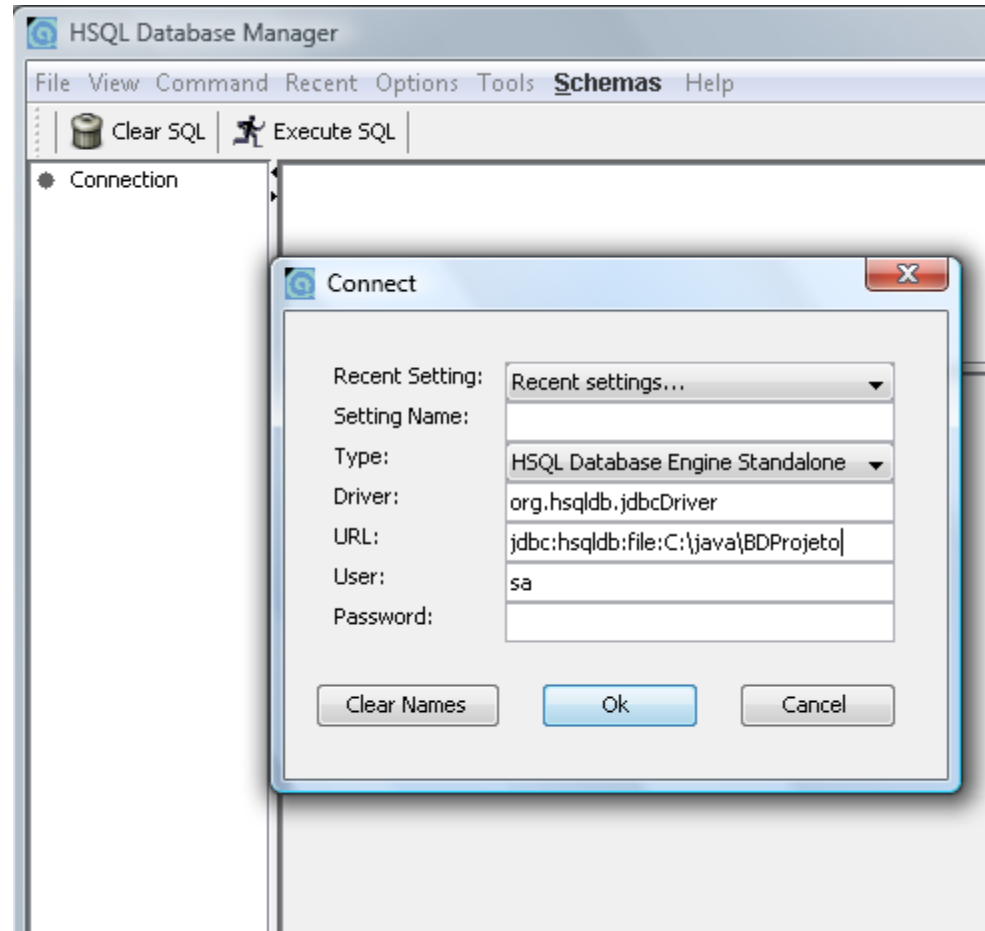
- ▶ Baixe o arquivo [hsqldb_1_8_0_10.zip](#)
- ▶ Descompacte-o
- ▶ Para não precisar ficar configurando em todo projeto do Netbeans que você quer utilizar o Hsqdb é interessante adicionar o caminho da pasta **lib** que está dentro da pasta onde você descompactou o Hsqdb, por exemplo:
 - ▶ Digite em uma janela DOS o comando:
`set CLASSPATH=%CLASSPATH%;C:\Java\Hsqldb\lib\hsqldb.jar`
 - ▶ Ou para que esta configuração fique definitiva: adicione às variáveis de ambiente do windows (aperte +BREAK, selecione a aba Avançado, depois clique no botão Variáveis de Ambiente, etc.)

Hsqldb – Criando um banco de dados

Aperte +R e digite: `java org.hsqldb.util.DatabaseManagerSwing`

Escolha o Tipo como na caixa de diálogo ao lado

Depois na URL digite a pasta e o nome do arquivo onde você deseja armazenar o seu banco



Hsqldb – Criando tabelas

```
CREATE TABLE CATEGORIA (  
    ID    IDENTITY PRIMARY KEY,  
    NOME  VARCHAR,  
    CONSTRAINT NOME_UNIQUE UNIQUE (NOME)  
)
```

Hsqldb – Criando tabelas

```
CREATE TABLE PRODUTO (  
    ID            IDENTITY PRIMARY KEY,  
    NOME          VARCHAR,  
    DESCRICAO     VARCHAR,  
    QTD_ESTOQUE  INTEGER,  
    CATEGORIA     INTEGER,  
    PRECO         DOUBLE,  
    CONSTRAINT FK_CATEGORIA FOREIGN KEY  
        (CATEGORIA) REFERENCES CATEGORIA (ID)  
)
```

Hsqldb – Criando tabelas

```
CREATE TABLE CLIENTE (  
    ID            IDENTITY PRIMARY KEY,  
    CPF           VARCHAR,  
    NOME          VARCHAR,  
    ENDERECO     VARCHAR,  
    TELEFONE     VARCHAR,  
    DT_NASC      DATE,  
    USERNAME     VARCHAR,  
    PASSWORD     VARCHAR,  
    CONSTRAINT CPF_UNIQUE UNIQUE (CPF)  
)
```


Hsqldb – Criando tabelas

```
CREATE TABLE CARRINHO (  
    ID          IDENTITY PRIMARY KEY,  
    CLIENTE    INTEGER,  
    DATA      DATE,  
    CONSTRAINT FK_CLIENTE FOREIGN KEY  
        (CLIENTE) REFERENCES CLIENTE (ID)  
)
```

Hsqldb – Criando tabelas

```
CREATE TABLE ITEM (  
    ID          IDENTITY PRIMARY KEY,  
    QTD         INTEGER,  
    PRODUTO     INTEGER,  
    CARRINHO    INTEGER,  
    CONSTRAINT FK_PRODUTO FOREIGN KEY  
        (PRODUTO) REFERENCES PRODUTO (ID) ,  
    CONSTRAINT FK_CARRINHO FOREIGN KEY  
        (CARRINHO) REFERENCES CARRINHO (ID)  
)
```

Hsqldb – Inserindo dados

```
INSERT INTO CLIENTE (CPF, NOME,  
    ENDERECO, TELEFONE, DT_NASC, USERNAME,  
    PASSWORD)
```

```
VALUES
```

```
('1', 'João dos Santos',  
'Rua das Pérolas, 222', '2222-3333',  
'1995-12-31', 'joao', 'j040')
```

Hsqldb – Inserindo dados

```
INSERT INTO PRODUTO
```

```
  (NOME, DESCRICAO, QTD_ESTOQUE,  
   CATEGORIA, PRECO)
```

```
VALUES
```

```
 ('Telefone s/ fio', 'Telefone sem fio,  
utilizando frequência 900MHz.  
Fabricante: Multibriz. 1 ano de  
garantia. Cor: Preto / Prata.', 10, 1,  
129.99)
```



Relacionamento

Hsqldb – Recuperando dados

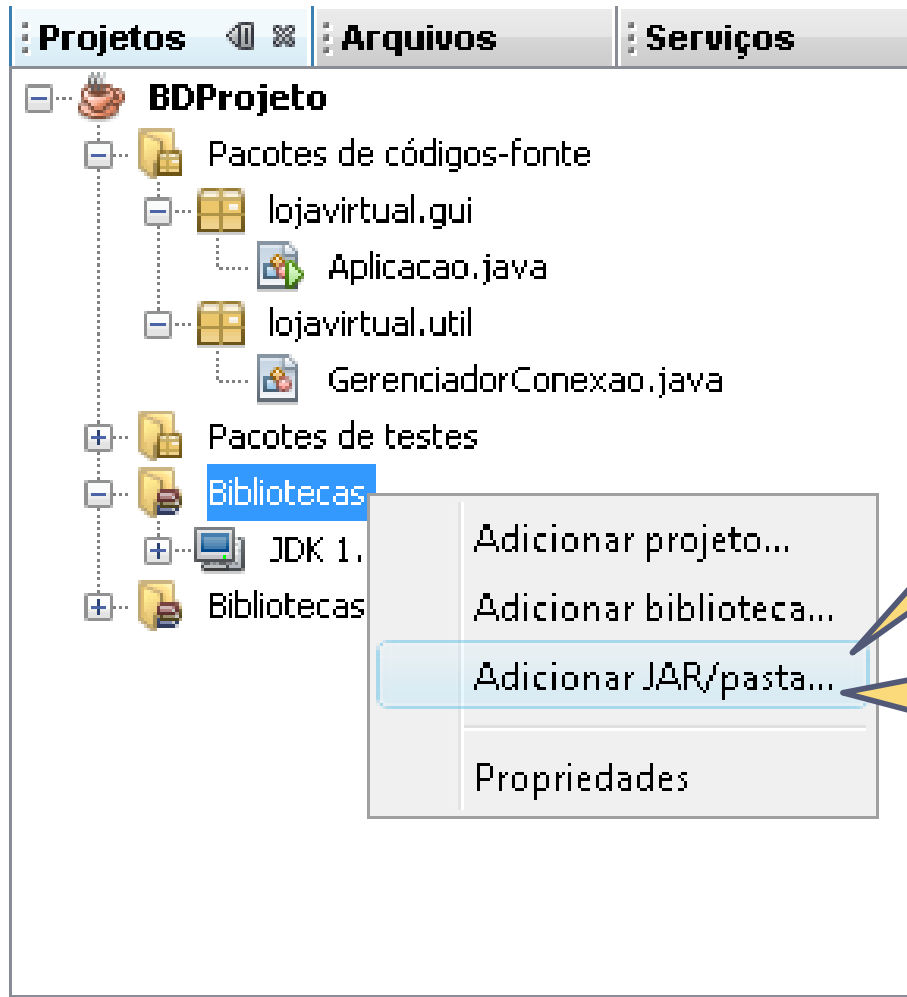
```
SELECT * FROM CATEGORIA
```

```
SELECT * FROM CATEGORIA WHERE ID = 1
```

```
SELECT * FROM CATEGORIA  
WHERE NOME = 'Telefonia'
```

ID	NOME
0	Eletrônicos
1	Telefonia

Netbeans – Reconhecendo Driver



Na próxima janela, selecione o arquivo `hsqldb.jar`

De preferência, copie antes o arquivo `hsqldb.jar` que está na pasta `lib`, para uma pasta `bd` dentro do seu projeto

JDBC – Criando conexão

```
private static Connection conexao = null;
private static boolean conectar() {
    try{
        Class.forName ("org.hsqldb.jdbcDriver");
        conexao = DriverManager.getConnection (
            "jdbc:hsqldb:file:c:/java/BDProjeto",
            "sa", "");
        return true;
    } catch (Exception ex) {
        return false;
    }
}
```

JDBC – Criando Statement

```
public static Statement criarStatement() {  
    Statement statement = null;  
    try {  
        if (conexao != null ||  
            GerenciadorConexao.conectar())  
            statement = conexao.createStatement();  
    } catch (SQLException ex) {  
    }  
    return statement;  
}
```


JDBC – Criando PreparedStatement

```
public static PreparedStatement
    criarPreparedStatement (String sql) {
    PreparedStatement prepareStatement = null;
    try {
        if (conexao != null ||
            GerenciadorConexao.conectar())
            prepareStatement =
                conexao.prepareStatement (sql);
    } catch (SQLException ex) {
    }
    return prepareStatement;
}
```

JDBC – Desconectando (Hsqldb)

```
public static void desconectar() {  
    Statement statement =  
        GerenciadorConexao.criarStatement();  
    if (statement != null) {  
        try {  
            statement.executeUpdate (  
                "SHUTDOWN COMPACT");  
        } catch (SQLException ex) {  
        } finally {  
            conexao = null;  
        }  
    }  
}
```

JDBC – Recuperando dados

```
Statement statement =
    GerenciadorConexao.criarStatement();
try {
    ResultSet rs = statement.executeQuery(
        "SELECT * FROM CLIENTE");
    while (rs.next()) {
        System.out.println(rs.getInt("ID") + "\t" +
            rs.getString("NOME"));
    }
    rs.close(); statement.close();
} catch (SQLException ex) {
    ex.printStackTrace();
}
```

JDBC – Recuperando dados

```
PreparedStatement preparedStatement =  
    GerenciadorConexao.criarPreparedStatement(  
        "SELECT * FROM CLIENTE WHERE NOME = ?");  
try {  
    preparedStatement.setString(1,  
        "João dos Santos");  
    ResultSet rs =  
        preparedStatement.executeQuery();  
    while (rs.next()) {  
        System.out.println(rs.getInt("ID") + "\t" +  
            rs.getString("NOME"));  
    }  
} catch (SQLException ex) {  
    ex.printStackTrace();  
}
```

Referencia a primeira
interrogação do comando
SQL acima