

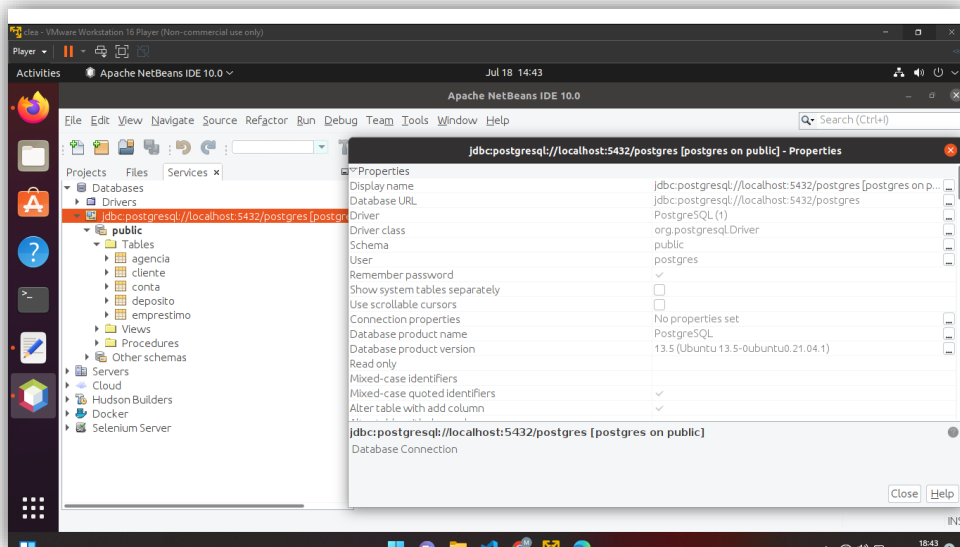
Sistemas de Bancos de Dados

Aula 08 – Estabelecendo conexões

Murielly Oliveira Nascimento – 11921BSI22

O objetivo desta aula é criarmos uma conexão do JDBC ao banco de dados postgres. Antes de iniciar, convém salientar que NetBeans também possui uma interface de conexão a bancos de dados similar à do programa pgadmin3. Caso você tenha instalado o programa NetBeans então poderá optar por usá-lo. Se você estiver com dificuldades para se conectar utilizando o pgadmin3 ou via o seu programa via JDBC uma opção é utilizar o NetBeans para tentar encontrar que tipo de erro de conexão você está enfrentando.

- 1) Na aba Serviços do NetBeans, opção Drivers, você tanto pode utilizar os drivers mapeados por padrão quanto se conectar a um novo banco de dados. No exemplo abaixo foram feitas duas conexões ao banco de testes do tutorial e uma conexão a um banco do SGBD PostgreSQL.

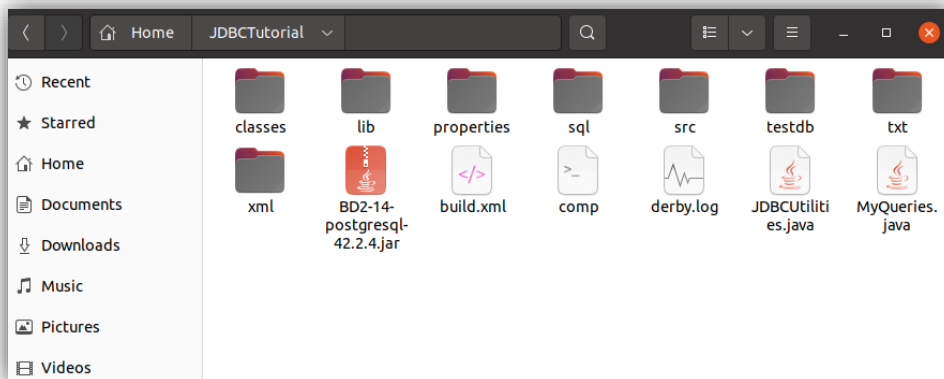


- 2) Baixe o arquivo postgresql-42.2.4.jar do moodle. Depois abra o terminal de comandos do linux e posicione-se no diretório em que baixou o arquivo postgresql-42.2.4.jar para esta aula. Para que não se recorde: use para isso o comando cd;

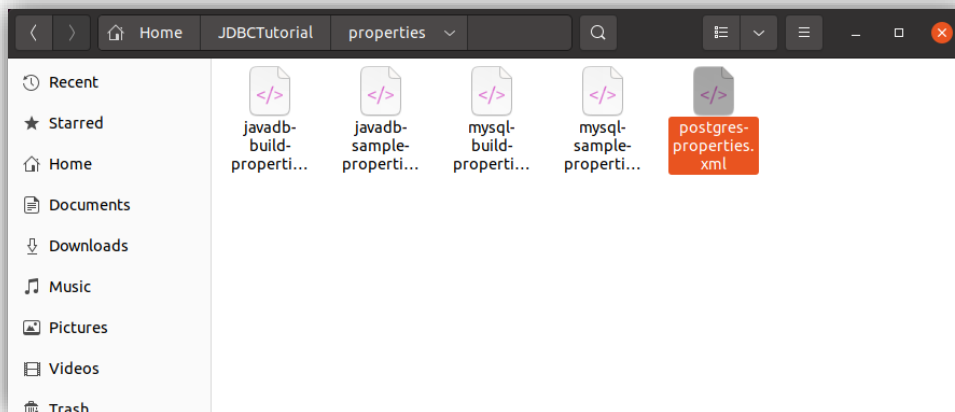
```
mury@ubuntu:~$ cd Downloads/  
mury@ubuntu:~/Downloads$ ls  
BD2-13-SBD07.pdf      BD2-14-postgresql-42.2.4.jar  SBD06c.pdf  
BD2-13-source_code  JDBCtutorial.zip  
mury@ubuntu:~/Downloads$
```

- 3) Copie o drive do postgres de nome postgresql-42.2.4.jar para a pasta raiz do tutorial do java digitando o comando cp abaixo:

```
mury@ubuntu:~/Downloads$ cp BD2-14-postgresql-42.2.4.jar /home/mury/JDBCTutorial/  
mury@ubuntu:~/Downloads$
```



- 4) Faça uma cópia do arquivo `mysql-sample-properties.xml`, que está na pasta `properties`, com o nome de `postgres-properties.xml`. Esse novo arquivo será utilizado para se conectar ao banco de dados do postgres. Terminada a cópia: edite o novo arquivo com os parâmetros corretos, lembre-se do usuário e senha do usuário postgres e da porta de conexão padrão. Se você não estiver conseguindo se lembrar, tente criar uma conexão usando, por exemplo, o NetBeans. Parâmetros iguais devem ser configurados no arquivo `postgres-properties.xml`.



```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
3 <properties>
4 |
5 <entry key="dbms">postgresql</entry>
6 <entry key="jar_file">/home/mury/JDBCTutorial/BD2-14-postgresql-42.2.4.jar</entry>
7 <entry key="driver">org.postgresql.Driver</entry>
8 <entry key="database_name">postgres</entry>
9 <entry key="user_name">postgres</entry>
10 <entry key="password">postgres</entry>
11 <entry key="server_name">localhost</entry>
12 <entry key="port_number">5432</entry>
13
14
15 </properties>

```

- 5) Modifique o programa JDBCUtilities.java para incorporar estas linhas que o habilitam a se conectar a um banco de dados postgres. Veja que no código existem duas funções com uma estrutura if para se conectar ao derby e ao mysql. Altere ambas as funções acrescentando este código:

```

String currentUrlString = null;

if (this.dbms.equals("mysql")) {
    currentUrlString = "jdbc:" + this.dbms + "://" + this.serverName + ":" + this.portNumber + "/" + this.dbName;
    conn = DriverManager.getConnection(currentUrlString, connectionProps);
    this.urlString = currentUrlString + this.dbName;
    conn.setCatalog(this.dbName);
} else if (this.dbms.equals("derby")) {
    this.urlString = "jdbc:" + this.dbms + ":" + this.dbName;
    conn = DriverManager.getConnection(this.urlString + ";create=true", connectionProps);
} else if (this.dbms.equals("postgresql")) {
    currentUrlString = "jdbc:" + this.dbms + "://" + this.serverName + ":" + this.portNumber + "/" + this.dbName;
    conn = DriverManager.getConnection(currentUrlString, connectionProps);
    this.urlString = currentUrlString + this.dbName;
    conn.setCatalog(this.dbName);
}

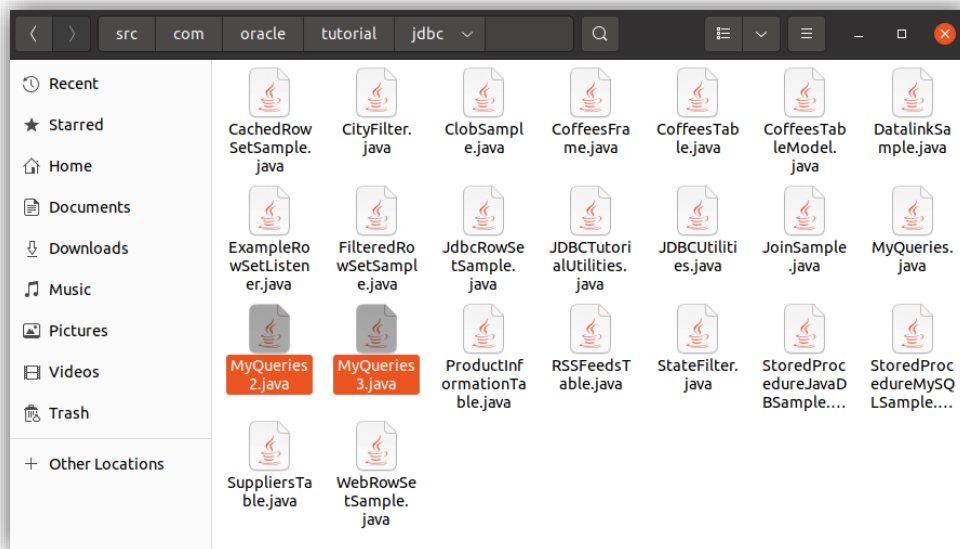
System.out.println("Connected to database");
return conn;

```

Perceba que em uma função a variável currentUrlString está declarada e na outra não. Na função sem declaração é necessário dizer qual é o tipo da variável ou teremos um erro de compilação. Acrescente o tipo String, antes da variável, na função na qual a variável não foi declarada.

- 6) Crie duas novas classes a partir da classe MyQueries.java (MyQueries2.java e MyQueries3.java) para realizar as seguintes consultas e exibir os resultados na tela da linha de comando. Para esse fim é necessário tanto mudar o nome do arquivo quanto das referências à classe que está definida dentro do arquivo ou a Java vai acusar erro

de compilação. Estas são as consultas que os novos programas devem fazer:



- a. Retorne os nomes de todos os clientes que possuem apenas depósitos, bem como a soma de depósitos de cada cliente (Implementar no MyQueries2.java);

```
public static void getMyData(Connection con) throws SQLException {
    Statement stmt = null;
    String query =
        "select d.nome_cliente, sum(d.saldo_deposito)" +
        "from deposito d where d.nome_cliente not in" +
        "(select e.nome_cliente from emprestimo e)" +
        "group by d.nome_cliente;";
}
```

- b. Retorne os nomes dos clientes que possuem depósitos e empréstimos (ambos) com as respectivas somas (Implementar no MyQueries3.java);

```
public static void getMyData(Connection con) throws SQLException {
    Statement stmt = null;
    String query =
        "select distinct d.nome_cliente, sum(d.saldo_deposito), sum(e.valor_emprestimo) " +
        "from deposito as d full outer join emprestimo as e " +
        "on d.nome_cliente = e.nome_cliente " +
        "where e.valor_emprestimo is not null " +
        "and d.saldo_deposito is not null " +
        "group by d.nome_cliente;";

    try {
        stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery(query);
        System.out.println("Soma de depositos e emprestimos dos clientes: ");
        while (rs.next()) {
            String Name = rs.getString(1);
            System.out.println("    " + Name);
            String deposito = rs.getString(2);
            System.out.println("        DEP - " + deposito);
            String emprestimo = rs.getString(3);
            System.out.println("        EMP - " + emprestimo);
        }
    }
}
```

- 7) Modifique o arquivo 'comp' da aula anterior para criar uma variável postgres com a configuração adequada à estrutura de diretórios de sua máquina. Use a variável derby como exemplo de como deve ficar a variável postgres dentro do arquivo comp.
- postgres=/home/myuser/mydir/JDBCTutorial/postgresql-42.2.4.jar
 - Acrescente a variável \$postgres no parâmetro -classpath (-cp) ao acionar o compilador java (denominado javac) e também ao acionar a máquina virtual java (denominado java). Modifique o script comp para este fim.
 - Lembre-se que se o arquivo comp foi copiado agora e ainda não utilizado então ele ainda não tem permissão de execução. Neste caso o comando chmod 755 comp vai habilitá-lo para este fim;

```
1 #!/bin/sh -
2
3 postgres=/home/mury/JDBCTutorial/BD2-14-postgresql-42.2.4.jar
4 mypath=/home/mury/JDBCTutorial
5 mypackage=com/oracle/tutorial/jdbc
6
7 if [ -r $mypath/src/$mypackage/$1.java -a -r $mypath/$2 ]
8 then
9
10     javac -cp "$postgres:$mypath/classes:$mypath/lib/JDBCTutorial.jar" $mypath/src/$mypackage/$1.java
11
12     if [ -r $mypath/src/$mypackage/$1.class ]
13     then
14         mv $mypath/src/$mypackage/$1.class $mypath/classes/$mypackage/
15
16         java -cp "$postgres:$mypath/classes" $mypackage/$1 $mypath/$2
17     fi
18 else
19     echo Missing file, check:
20     ls $mypath/src/$mypackage/$1.java
21     ls $mypath/$2
22 fi
```

- 8) Compile primeiro a classe JDBCUtilities.java que foi modificada nas etapas anteriores:
./comp JDBCUtilities properties/javadb-sample-properties.xml

```
mury@ubuntu:~/JDBCTutorial$ ./comp JDBCUtilities properties/postgres-properties.xml
Reading properties file /home/mury/JDBCTutorial/properties/postgres-properties.xml
Set the following properties:
dbms: postgresql
driver: org.postgresql.Driver
dbName: postgres
userName: postgres
serverName: localhost
portNumber: 5432
Connected to database
org.postgresql.jdbc.PgConnection@1ff8b8f
Releasing all open resources ...
mury@ubuntu:~/JDBCTutorial$
```

- 9) Compile e execute em seguida as duas classes MyQueries2 e MyQueries3.

Lembre-se que o novo arquivo postgres-properties.xml (passo 4) deve ser passado para todos os programas compilados e executados com o programa comp.

```
mury@ubuntu:~/JDBCTutorial$ ./comp MyQueries2 properties/postgres-properties.xml
Set the following properties:
dbms: postgresql
driver: org.postgresql.Driver
dbName: postgres
userName: postgres
serverName: localhost
portNumber: 5432
Connected to database
org.postgresql.jdbc.PgConnection@387c703b
Soma dos depositos dos clientes
  Cláudia Santos Mota      3599.29
  Thiago Andrade Fiuza    479.66
  Wantuil Diniz e Souza    84.17
  Alexandre Marcio de Souza 1201.37
  Felipe Peres Bichara Junior 7417.87
Releasing all open resources ...
mury@ubuntu:~/JDBCTutorial$
```

```
mury@ubuntu:~/JDBCTutorial$ ./comp MyQueries3 properties/postgres-properties.xml
Set the following properties:
dbms: postgresql
driver: org.postgresql.Driver
dbName: postgres
userName: postgres
serverName: localhost
portNumber: 5432
Connected to database
org.postgresql.jdbc.PgConnection@387c703b
Soma de depositos e emprestimos dos clientes:
  Marcos Andrade
    DEP - 3493.63
    EMP - 886.85
  Germano Luiz de Paula
    DEP - 9635.44
    EMP - 5026.95
  Gilmar Negreiros Carvalho
    DEP - 10230.12
    EMP - 4234.1
  Everardo Monfort Leitão
    DEP - 3562.54
    EMP - 3773.96
```