

EULLER HENRIQUE BANDEIRA OLIVEIRA
11821BSI210

T06-Estabelecendo conexões

1) 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;

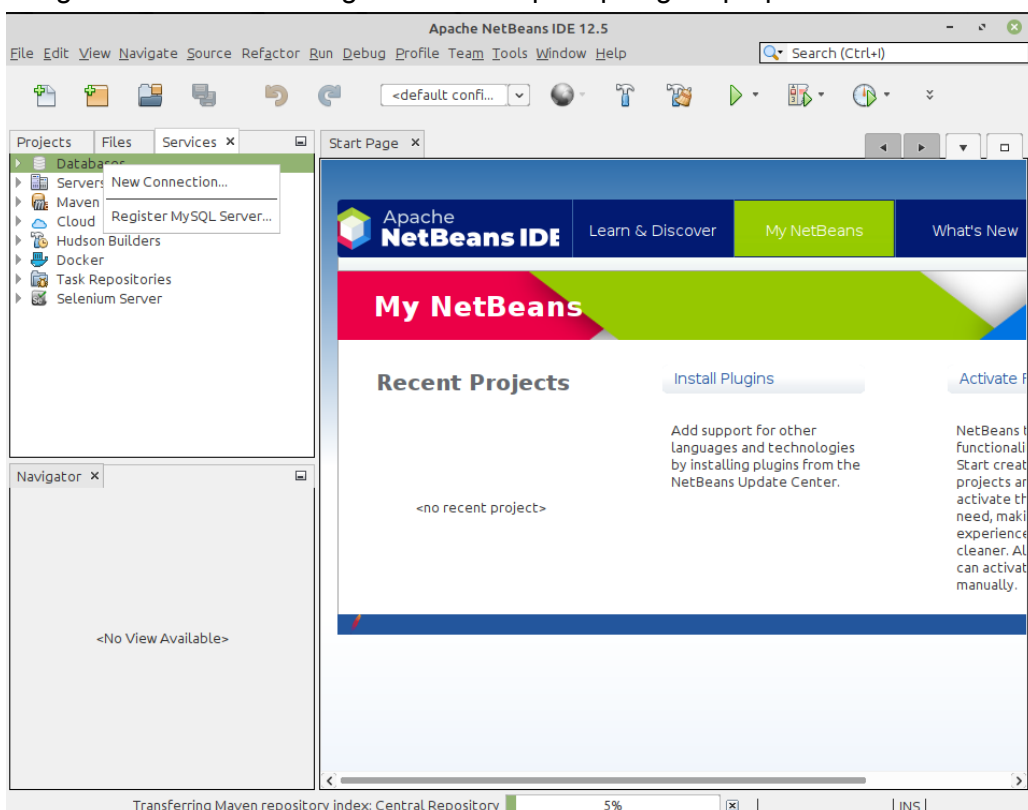
```
euller@euller-VirtualBox: ~/Downloads
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
euller@euller-VirtualBox:~/Downloads$ ls
BD2-14-postgresql-42.2.4.jar  JDBCtutorial.zip  oi.txt  sample.tar.gz
IB.dump                      oi.tar.gz        sample
euller@euller-VirtualBox:~/Downloads$
```

2) 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:

1. cp postgresql-42.2.4.jar /home/myuser/mydir/JDBCTutorial/

```
euller@euller-VirtualBox: ~/Documentos/mydir/JDBCTutorial
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
euller@euller-VirtualBox:~/Downloads$ cp BD2-14-postgresql-42.2.4.jar /home/euller/Documentos/mydir/JDBCTutorial/
euller@euller-VirtualBox:~/Downloads$ cd /home/euller/Documentos/mydir/JDBCTutorial/
euller@euller-VirtualBox:~/Documentos/mydir/JDBCTutorial$ ls
BD2-14-postgresql-42.2.4.jar  comp  lib  sql  txt
build.xml                  derby.log  MyQueries.java  src  xml
classes                    JDBCUtilities.java  properties  testdb
euller@euller-VirtualBox:~/Documentos/mydir/JDBCTutorial$
```

3) 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.



New Connection Wizard

Locate Driver

Driver: PostgreSQL
Driver File(s): /home/euller/netbeans-12.5/netbeans/ide/modules/ext/pos
Add...
Remove

Back
Next >
Finish
Cancel
Help

Customize Connection

Driver Name: PostgreSQL
Host: localhost
Port: 5432
Database: postgres
User Name: postgres
Password:
Remember password
Connection Properties
Test Connection
JDBC URL: jdbc:postgresql://localhost:5432/postgres

Connection Succeeded.
Back
Next >
Finish
Cancel
Help

New Connection Wizard

Choose name for connection

Override the default name for the connection. The name should be descriptive about the connection you are creating.
Input connection name:
jdbc:postgresql://localhost:5432/postgres [postgres on Default schema]

Back
Next >
Finish
Cancel
Help

New Connection Wizard

Choose Database Schema

For each database connection, the Services window only displays objects from one database schema. Select the schema of the tables to be displayed.
Select schema: information_schema

Back
Next >
Finish
Cancel
Help

Apache NetBeans IDE 12.5

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects Files Services

Databases
Java DB
Drivers
jdbc:postgresql://localhost:5432/postgres [postgres on information_schema]
Information_schema
Other schemas
Servers
Maven Repositories
Cloud
Hudson Builders
Docker
Task Repositories
Selenium Server

Navigator

No View Available

Start Page

Properties

Display name: jdbc:postgresql://localhost:5432/postgres [pos...
Database URL: jdbc:postgresql://localhost:5432/postgres
Driver: Postgres
Driver class: org.postgresql.Driver
Schema: information_schema
User: postgres
Remember password:
Show system tables separately:
Use scrollable cursors:
Connection properties: No properties set
Database product name: PostgreSQL
Database product version: 12.8 (Ubuntu 12.8-0ubuntu0.20.04.1)
Read only:
Mixed-case identifiers:
Mixed-case quoted identifiers:
Alter table with add column:
Alter table with drop column:
Convert:
Table correlation names:
Expressions in order by:
Unrelated order by:
Group by support:
Unrelated group by support:
Group by beyond select:

jdbc:postgresql://localhost:5432/postgres [postgres on information_schema]

Database Connection

Unpacking Index for Central Repository

INS

```

euller@euller-VirtualBox: ~/Documentos/mydir/JDBCTutorial/properties
Arquivo Editar Ver Pesquisar Terminal Ajuda
GNU nano 4.8 postgres-properties.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>

  <entry key="dbms">postgresql</entry>
  <entry key="driver">org.postgresql.Driver</entry>
  <entry key="database_name">IB</entry>
  <entry key="user_name">postgres</entry>
  <entry key="password">postgres</entry>
  <entry key="server_name">localhost</entry>
  <entry key="port_number">5432</entry>

</properties>

Escritas 14 linhas
^G Obter Ajuda ^O Gravar ^W Onde está? ^K Recort txt ^J Justificar ^C Pos atual
^X Sair ^R Ler o arq ^S Substituir ^U Colar txt ^T Verifortog ^_ Ir p/ linha

```

4) 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:

```
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);
}
```

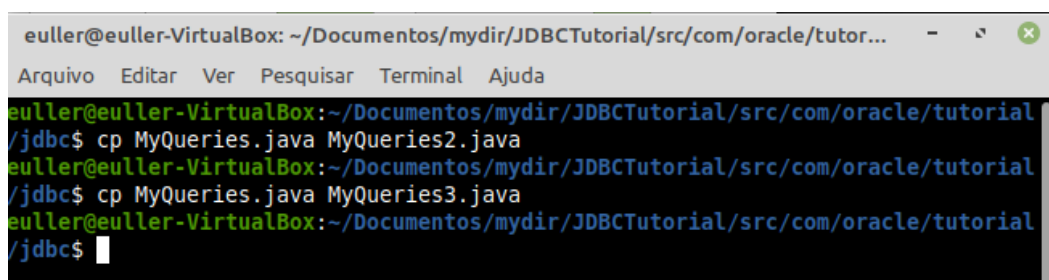
Perceba que em uma função a variável currentUrlString está declarada e na outra não. Na função sem declaração é necessário dizer qual é o tipo da variável ou teremos um erro de compilação. Acrescente o tipo String, antes da variável, na função na qual a variável não foi declarada

```
public Connection getConnection() throws SQLException {
    Connection conn = null;
    Properties connectionProps = new Properties();
    connectionProps.put("user", this.userName);
    connectionProps.put("password", this.password);

    String currentUrlString = null;

    if (this.dbms.equals("mysql")) {
        currentUrlString = "jdbc:" + this.dbms + "://" + this.serverName + ":" + this.portNumber + "/" +
        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;
}
```

5) 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:



```
euller@euller-VirtualBox: ~/Documentos/mydir/JDBCTutorial/src/com/oracle/tutor...
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
euller@euller-VirtualBox:~/Documentos/mydir/JDBCTutorial/src/com/oracle/tutorial
/jdbc$ cp MyQueries.java MyQueries2.java
euller@euller-VirtualBox:~/Documentos/mydir/JDBCTutorial/src/com/oracle/tutorial
/jdbc$ cp MyQueries.java MyQueries3.java
euller@euller-VirtualBox:~/Documentos/mydir/JDBCTutorial/src/com/oracle/tutorial
/jdbc$
```

1. 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;
    /* Retorne os nomes de todos os clientes que possuem apenas depósitos, bem como a soma de depósitos de cada cliente */
    String query =
        "SELECT nome_cliente, sum(saldo_deposito) FROM DEPOSITO WHERE nome_cliente NOT IN (SELECT nome_cliente FROM EMPRESTIMO) GROUP BY nome_cliente";

    try {
        stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery(query);
        while (rs.next()) {
            String name = rs.getString(1);
            String saldo = rs.getString(2);
            System.out.println(" " + name + " " + saldo);
        }
    } catch (SQLException e) {
        JDBCUtilities.printSQLException(e);
    } finally {
        if (stmt != null) { stmt.close(); }
    }
}
```

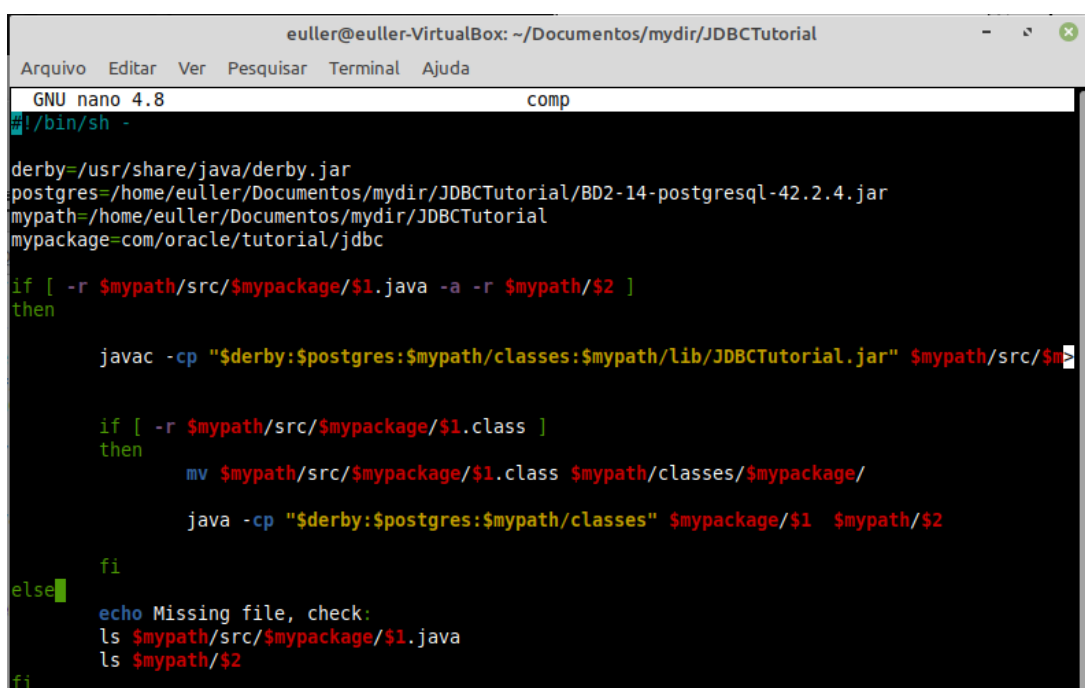
2. Retorne os nomes dos clientes que possuem depósitos e empréstimos (ambos) com as respectivas somas (Implementar no MyQueries3.java);

```
public static void getMyData(Connection con) throws SQLException {
    Statement stmt = null;
    /* Retorne os nomes dos clientes que possuem depósitos e empréstimos (ambos) com as respectivas somas */
    String query =
        "SELECT nome_cliente, COALESCE(SUM(saldo_deposito),0) AS soma FROM deposito WHERE nome_cliente IN (SELECT nome_cliente FROM EMPRESTIMO) GROUP BY nome_cliente UNION SELECT "
        "nome_cliente, -1 * COALESCE(SUM(valor_emprestimo),0) AS soma FROM emprestimo WHERE nome_cliente IN (SELECT nome_cliente FROM DEPOSITO) GROUP BY nome_cliente";

    try {
        stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery(query);
        while (rs.next()) {
            String name = rs.getString(1);
            String saldo = rs.getString(2);
            System.out.println(" " + name + " " + saldo);
        }
    } catch (SQLException e) {
        JDBCUtilities.printSQLException(e);
    } finally {
        if (stmt != null) { stmt.close(); }
    }
}
```

6) 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.

1. postgres=/home/myuser/mydir/JDBCTutorial/postgresql-42.2.4.jar
2. 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.



```
euller@euller-VirtualBox: ~/Documentos/mydir/JDBCTutorial
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda

GNU nano 4.8                                comp
#!/bin/sh -

derby=/usr/share/java/derby.jar
postgres=/home/euller/Documentos/mydir/JDBCTutorial/BD2-14-postgresql-42.2.4.jar
mypath=/home/euller/Documentos/mydir/JDBCTutorial
mypackage=com/oracle/tutorial/jdbc

if [ -r $mypath/src/$mypackage/$1.java -a -r $mypath/$2 ]
then

    javac -cp "$derby:$postgres:$mypath/classes:$mypath/lib/JDBCTutorial.jar" $mypath/src/$1.java

    if [ -r $mypath/src/$mypackage/$1.class ]
    then
        mv $mypath/src/$mypackage/$1.class $mypath/classes/$mypackage/
        java -cp "$derby:$postgres:$mypath/classes" $mypackage/$1 $mypath/$2
    fi
else
    echo Missing file, check:
    ls $mypath/src/$mypackage/$1.java
    ls $mypath/$2
fi
```

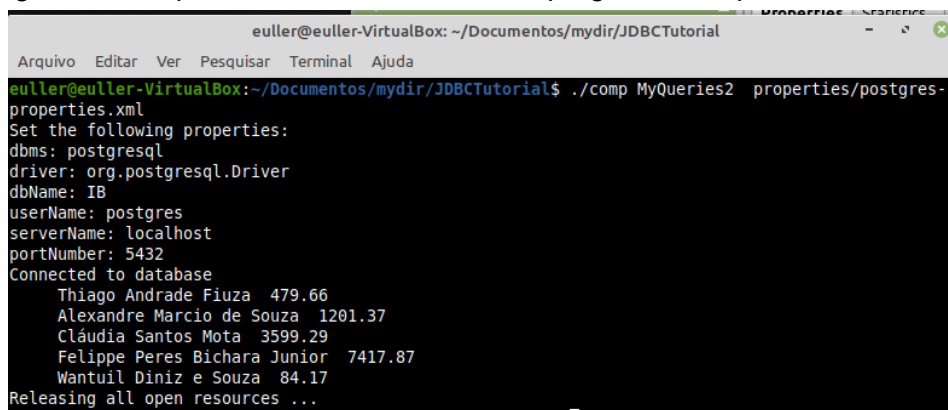
3. Lembre-se que se o arquivo comp foi copiado agora e ainda não utilizado então ele ainda não tem permissão de execução. Neste caso o comando `chmod 755 comp` vai habilitá-lo para este fim;

```
euller@euller-VirtualBox:~/Documentos/mydir/JDBCTutorial$ chmod 755 comp
```

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

```
euller@euller-VirtualBox:~/Documentos/mydir/JDBCTutorial$ ./comp JDBCUtilities properties/javadb-sample-properties.xml
Reading properties file /home/euller/Documentos/mydir/JDBCTutorial/properties/javadb-sample-properties.xml
Set the following properties:
dbms: derby
driver: org.apache.derby.jdbc.EmbeddedDriver
dbName: testdb
userName:
serverName: localhost
portNumber: 3306
Connected to database
Releasing all open resources ...
```

8) 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`



```
euller@euller-VirtualBox:~/Documentos/mydir/JDBCTutorial$ ./comp MyQueries2 properties/postgres-properties.xml
Set the following properties:
dbms: postgresql
driver: org.postgresql.Driver
dbName: IB
userName: postgres
serverName: localhost
portNumber: 5432
Connected to database
Thiago Andrade Fiuza 479.66
Alexandre Marcio de Souza 1201.37
Cláudia Santos Mota 3599.29
Felippe Peres Bichara Junior 7417.87
Wantuil Diniz e Souza 84.17
Releasing all open resources ...
```

```
euller@euller-VirtualBox:~/Documentos/mydir/JDBCTutorial$ ./comp MyQueries3 properties/postgres-properties.xml
Set the following properties:
dbms: postgresql
driver: org.postgresql.Driver
dbName: IB
userName: postgres
serverName: localhost
portNumber: 5432
Connected to database
Everardo Monfort Leitão 3562.54
Carolina Soares -1324.74
Clayton Pereira Bonfim 3231.84
Joaquim Carlos Reis -805.88
Gilmar Negreiros Carvalho -2117.05
Carolina Soares 3516.28
Marcos Andrade 3493.63
Bruno Miranda Pacheco de Castro -1139.08
Everardo Monfort Leitão -1886.98
Thiago Leôncio Guimarães -1411.99
Marcos Andrade -886.85
Bruno Miranda Pacheco de Castro 2512.88
Andre Cabral da Silva -2612.98
Andre Cabral da Silva 4490.47
Joaquim Carlos Reis 498.56
Germano Luiz de Paula 4817.72
Thiago Leôncio Guimarães 1753.42
Marco Aurélio Santos -3775.6399999999994
Marco Aurélio Santos 3164.44
Clayton Pereira Bonfim -896.28
Germano Luiz de Paula -1675.65
Gilmar Negreiros Carvalho 5115.06
Releasing all open resources ...
euller@euller-VirtualBox:~/Documentos/mydir/JDBCTutorial$
```