

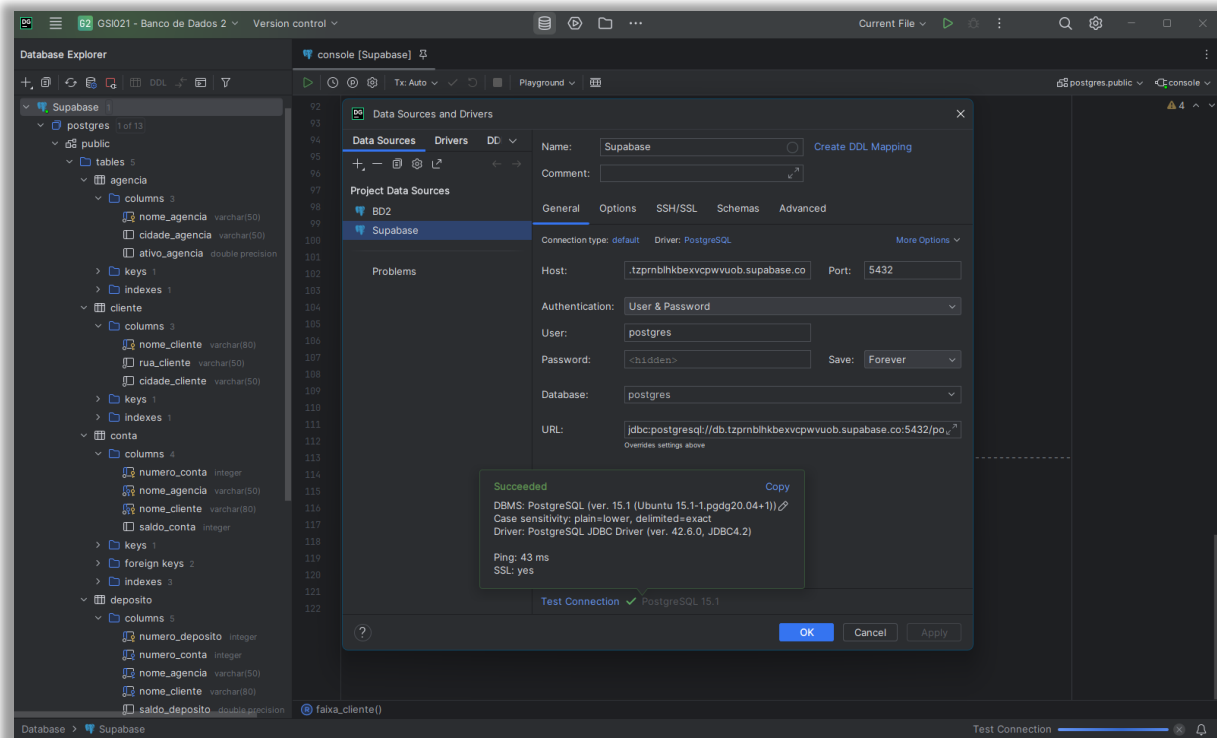
T05 – Estabelecendo conexões

Igor Augusto Reis Gomes – 12011BSI290 – igor.augusto@ufu.br

Heitor Guimarães Da Fonseca Filho – 12011BSI203 – heitor.filho@ufu.br

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

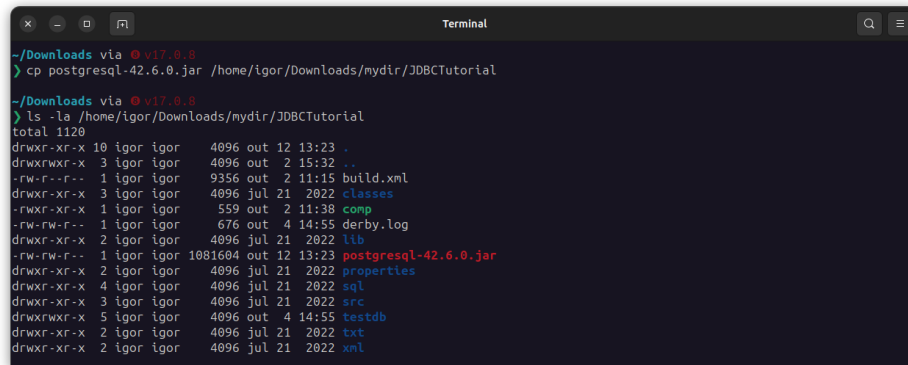
Utilizei o Datagrip:



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;

```
Terminal
~/Downloads
> cd Downloads
~/Downloads via v17.0.8
> ls -la
total 1244
drwxr-xr-x 7 igor igor 4096 out 12 13:19 .
drwxr-xr-x 35 igor igor 4096 out 9 15:45 ..
drwxr-xr-x 6 igor igor 4096 set 4 11:14 BD2
-rw-rw-r-- 1 igor igor 135496 out 12 12:25 BD2-14-SBD08.pdf
drwxr-xr-x 4 igor igor 4096 set 4 11:37 .BD2-main
-rw-rw-r-- 1 igor igor 18707 set 14 14:24 IB.dump
drwxrwxr-x 3 igor igor 4096 out 2 15:32 mydir
-rw-rw-r-- 1 igor igor 1081604 out 12 13:04 postgresql-42.6.0.jar
drwxrwxr-x 4 igor igor 4096 set 14 00:00 sample
drwxr-xr-x 4 igor igor 4096 set 12 11:10 Trabalhos
~/Downloads via v17.0.8
>
```

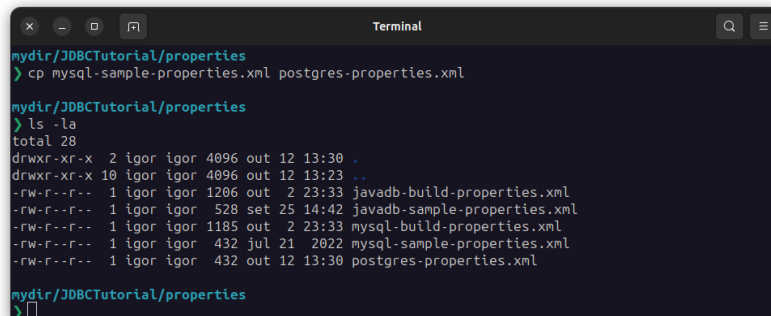
3. Copie o driver do postgres de nome postgresql-42.2.4.jar para a pasta raiz do tutorial do java digitando o comando cp: cp postgresql-42.2.4.jar /home/myuser/mydir/JDBCTutorial/



```
~/Downloads via v17.0.0
> cp postgresql-42.6.0.jar /home/igor/Downloads/mydir/JDBCTutorial

~/Downloads via v17.0.0
> ls -la /home/igor/Downloads/mydir/JDBCTutorial
total 1120
drwxr-xr-x 10 igor igor 4096 out 12 13:23 .
drwxr-xr-x  3 igor igor 4096 out 2 15:32 ..
-rw-r--r--  1 igor igor 9356 out 2 11:15 build.xml
drwxr-xr-x  3 igor igor 4096 jul 21 2022 classes
-rwxr-xr-x  1 igor igor 559 out 2 11:38 comp
-rw-rw-r--  1 igor igor 676 out 4 14:55 derby.log
drwxr-xr-x  2 igor igor 4096 jul 21 2022 lib
-rw-rw-r--  1 igor igor 1081604 out 12 13:23 postgresql-42.6.0.jar
drwxr-xr-x  2 igor igor 4096 jul 21 2022 properties
drwxr-xr-x  4 igor igor 4096 jul 21 2022 sql
drwxr-xr-x  3 igor igor 4096 jul 21 2022 src
drwxr-xr-x  5 igor igor 4096 out 4 14:55 testdb
drwxr-xr-x  2 igor igor 4096 jul 21 2022 txt
drwxr-xr-x  2 igor igor 4096 jul 21 2022 xml
```

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.

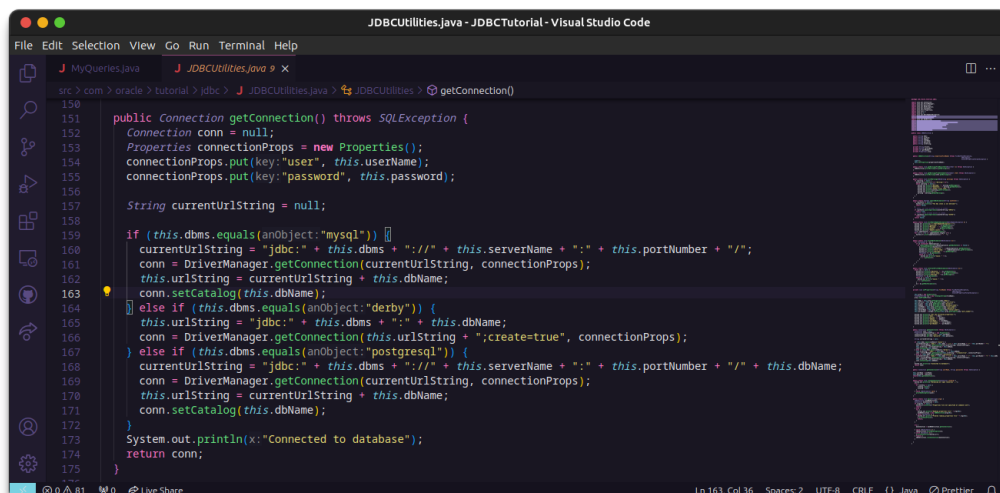


```
mydir/JDBCTutorial/properties
> cp mysql-sample-properties.xml postgres-properties.xml

mydir/JDBCTutorial/properties
> ls -la
total 28
drwxr-xr-x  2 igor igor 4096 out 12 13:30 .
drwxr-xr-x 10 igor igor 4096 out 12 13:23 ..
-rw-r--r--  1 igor igor 1206 out 2 23:33 javadb-build-properties.xml
-rw-r--r--  1 igor igor 528 set 25 14:42 javadb-sample-properties.xml
-rw-r--r--  1 igor igor 1185 out 2 23:33 mysql-build-properties.xml
-rw-r--r--  1 igor igor 432 jul 21 2022 mysql-sample-properties.xml
-rw-r--r--  1 igor igor 432 out 12 13:30 postgres-properties.xml

mydir/JDBCTutorial/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:



```
JDBCUtilities.java - JDBCTutorial - Visual Studio Code

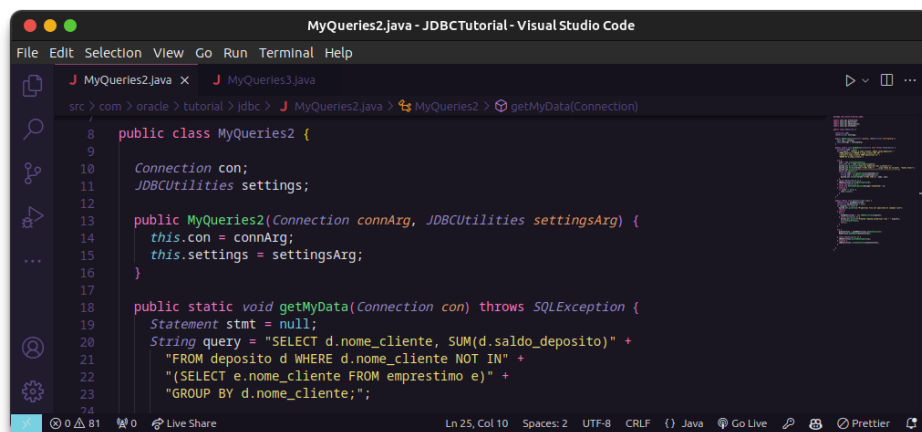
src > com > oracle > tutorial > jdbc > JDBCTutorial > JDBCUtilities > getConnection()

150
151 public Connection getConnection() throws SQLException {
152     Connection conn = null;
153     Properties connectionProps = new Properties();
154     connectionProps.put(key:"user", this.userName);
155     connectionProps.put(key:"password", this.password);
156
157     String currentUrlString = null;
158
159     if (this.dbms.equals(anObject:"mysql")) {
160         currentUrlString = "jdbc:" + this.dbms + "://" + this.serverName + ":" + this.portNumber + "/" + this.dbName;
161         conn = DriverManager.getConnection(currentUrlString, connectionProps);
162         this.urlString = currentUrlString + this.dbName;
163         conn.setCatalog(this.dbName);
164     }
165     else if (this.dbms.equals(anObject:"derby")) {
166         this.urlString = "jdbc:" + this.dbms + "://" + this.dbName;
167         conn = DriverManager.getConnection(this.urlString + ";create=true", connectionProps);
168     }
169     else if (this.dbms.equals(anObject:"postgresql")) {
170         currentUrlString = "jdbc:" + this.dbms + "://" + this.serverName + ":" + this.portNumber + "/" + this.dbName;
171         conn = DriverManager.getConnection(currentUrlString, connectionProps);
172         this.urlString = currentUrlString + this.dbName;
173         conn.setCatalog(this.dbName);
174     }
175     System.out.println(x:"Connected to database");
176     return conn;
177 }
```

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:

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

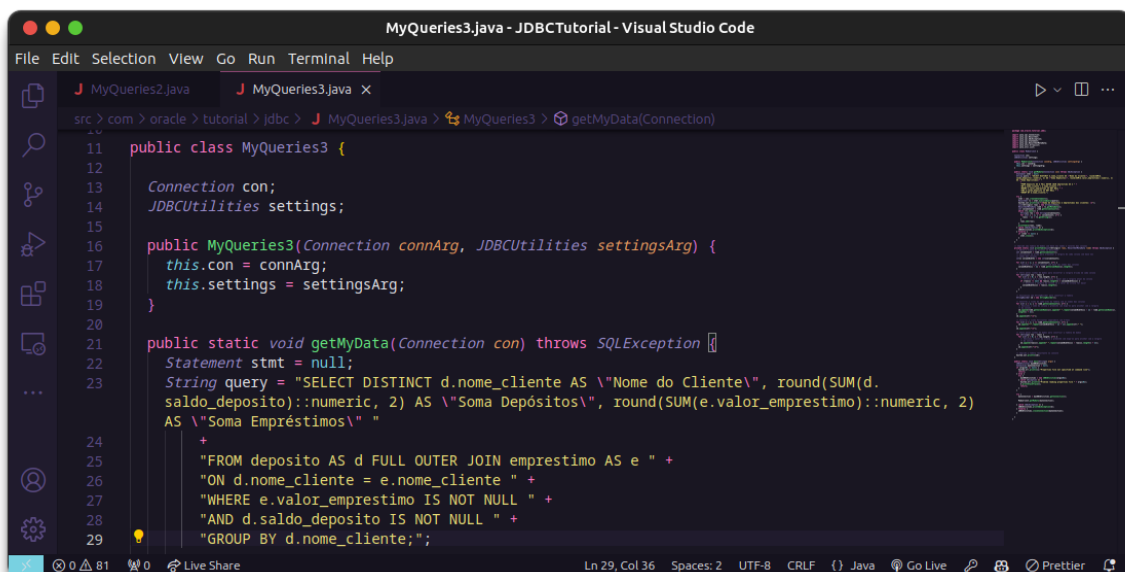
Nessa consulta, o subselect retorna os nomes dos clientes que possuem empréstimos e o select principal retorna os nomes dos clientes que não estão na lista retornada pelo subselect.



```
MyQueries2.java - JDBC Tutorial - Visual Studio Code
File Edit Selection View Go Run Terminal Help
src > com > oracle > tutorial > jdbc > MyQueries2.java > MyQueries2 > getMyData(Connection)
7
8 public class MyQueries2 {
9
10     Connection con;
11     JDBCUtilities settings;
12
13     public MyQueries2(Connection connArg, JDBCUtilities settingsArg) {
14         this.con = connArg;
15         this.settings = settingsArg;
16     }
17
18     public static void getMyData(Connection con) throws SQLException {
19         Statement stmt = null;
20         String query = "SELECT d.nome_cliente, SUM(d.saldo_deposito)" +
21             "FROM deposito d WHERE d.nome_cliente NOT IN " +
22             "(SELECT e.nome_cliente FROM emprestimo e)" +
23             "GROUP BY d.nome_cliente;";
24     }
25 }
```

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

Já nessa consulta, utilizei “Distinct” para não repetir os nomes dos clientes e o “full outer join” para que apareçam os clientes que possuem apenas depósitos ou apenas empréstimos. Adicionalmente, também utilizei alias para os nomes das colunas, e fiz uso do “round” para arredondar os valores (para duas casas) em conjunto com o “::numeric” para converter os valores para o tipo “numeric” (para que o round funcione).



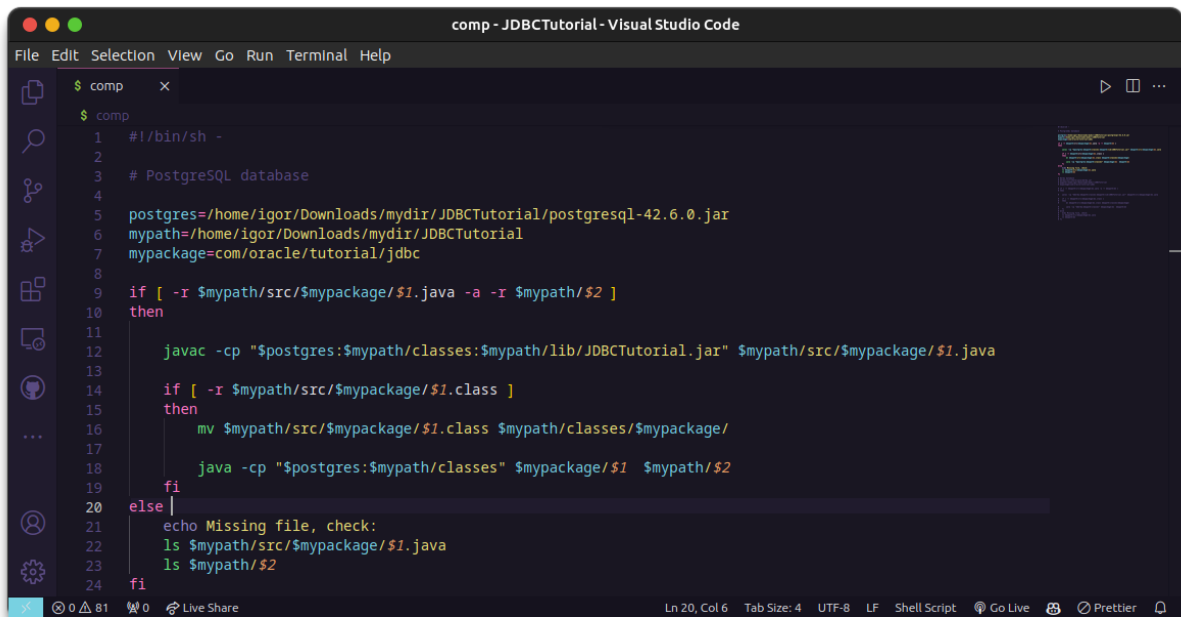
```
MyQueries3.java - JDBC Tutorial - Visual Studio Code
File Edit Selection View Go Run Terminal Help
src > com > oracle > tutorial > jdbc > MyQueries3.java > MyQueries3 > getMyData(Connection)
11 public class MyQueries3 {
12
13     Connection con;
14     JDBCUtilities settings;
15
16     public MyQueries3(Connection connArg, JDBCUtilities settingsArg) {
17         this.con = connArg;
18         this.settings = settingsArg;
19     }
20
21     public static void getMyData(Connection con) throws SQLException {
22         Statement stmt = null;
23         String query = "SELECT DISTINCT d.nome_cliente AS \"Nome do Cliente\", round(SUM(d." +
24             "saldo_deposito)::numeric, 2) AS \"Soma Depósitos\", round(SUM(e.valor_emprestimo)::numeric, 2) " +
25             "AS \"Soma Empréstimos\" " +
26             "FROM deposito AS d FULL OUTER JOIN emprestimo AS e " +
27             "ON d.nome_cliente = e.nome_cliente " +
28             "WHERE e.valor_emprestimo IS NOT NULL " +
29             "AND d.saldo_deposito IS NOT NULL " +
30             "GROUP BY d.nome_cliente;";
31     }
32 }
```

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.

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

7.2. 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.

7.3. 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;

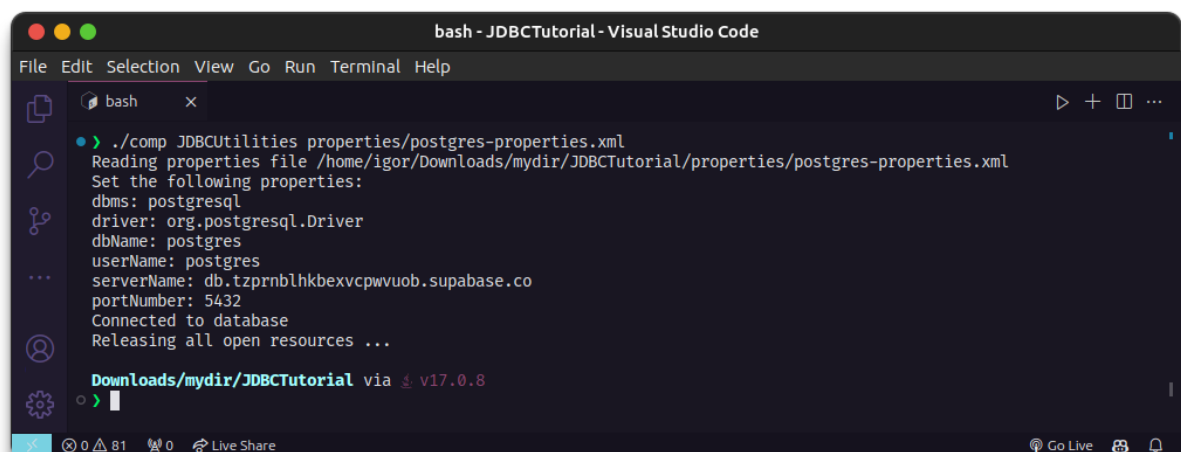


```
comp - JDBCTutorial - Visual Studio Code
File Edit Selection View Go Run Terminal Help

$ comp
$ comp
1  #!/bin/sh -
2
3  # PostgreSQL database
4
5  postgres=/home/igor/Downloads/mydir/JDBCTutorial/postgresql-42.6.0.jar
6  mypath=/home/igor/Downloads/mydir/JDBCTutorial
7  mypackage=com/oracle/tutorial/jdbc
8
9  if [ -r $mypath/src/$mypackage/$1.java -a -r $mypath/$2 ]
10 then
11
12     javac -cp "$postgres:$mypath/classes:$mypath/lib/JDBCTutorial.jar" $mypath/src/$mypackage/$1.java
13
14     if [ -r $mypath/src/$mypackage/$1.class ]
15     then
16         mv $mypath/src/$mypackage/$1.class $mypath/classes/$mypackage/
17     fi
18     java -cp "$postgres:$mypath/classes" $mypackage/$1 $mypath/$2
19 fi
20 else
21     echo Missing file, check:
22     ls $mypath/src/$mypackage/$1.java
23     ls $mypath/$2
24 fi
```

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

Utilizei como banco o serviço [Supabase](https://supabase.com), uma alternativa que achei mais adequada ao banco local:



```
bash - JDBCTutorial - Visual Studio Code
File Edit Selection View Go Run Terminal Help

bash
> ./comp JDBCUtilities properties/postgres-properties.xml
Reading properties file /home/igor/Downloads/mydir/JDBCTutorial/properties/postgres-properties.xml
Set the following properties:
dbms: postgresql
driver: org.postgresql.Driver
dbName: postgres
userName: postgres
serverName: db.tzprnblhkbexvcpwvuob.supabase.co
portNumber: 5432
Connected to database
Releasing all open resources ...

Downloads/mydir/JDBCTutorial via v17.0.8
>
```

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.

Fiz uma tabela básica na primeira consulta, e uma mais elaborada (com uma função) na segunda:

The screenshot shows the Visual Studio Code interface with the JDBC Tutorial project. The left pane displays the output of a Java program that connects to a PostgreSQL database and prints the sum of deposits and loans for clients. The right pane displays the output of a Java program that connects to a PostgreSQL database and prints the sum of deposits and loans for clients. The output is a table with columns: Nome do Cliente, Saldo Total, Soma Depósitos, and Soma Empréstimos. The table lists 10 clients and their respective values.

Nome do Cliente	Saldo Total	Soma Depósitos	Soma Empréstimos
Cláudia Santos Mota	3599.29	8980.94	5225.96
Thiago Andrade Fiuza	479.66	2512.88	1139.08
Wantuil Diniz e Souza	84.17	3516.28	1324.74
Alexandre Marcio de Souza	1201.37	3231.84	1792.56
Felippe Peres Bichara Junior	7417.87	3562.54	3773.96
		209635.44	167570026.95
		10230.12	4234.10
		997.12	805.88
		9493.32	3775.64
		3493.63	886.85
		3506.84	1411.99