

Railson Da Silva Martins - 11811BSI208

Wothon Mateus de Araujo - 12111BSI262

1. Download e cópia de arquivos:

Baixado o código fonte das classes **MyQueries.java** e **JDBCUtilities.java** para o diretório raiz do tutorial e realizado cópia dos arquivos para pastas solicitadas.

```

railson@railson-virtual-machine:~/mydir/JDBCTutorial$ cp MyQueries.java src/com/oracle/tutorial/jdbc/
railson@railson-virtual-machine:~/mydir/JDBCTutorial$ cp JDBCUtilities.java src/com/oracle/tutorial/jdbc/
railson@railson-virtual-machine:~/mydir/JDBCTutorial$

```

2. Inclusão de sessão:

Realizada inclusão de sessão no arquivo **build.xml** para execução da classe **MyQueries.java**.

Executado comando **sudo gedit build.xml** para abertura e edição de arquivo e incluído o seguinte código:

```

40 <target name="mycode">
41 <java classname="com.oracle.tutorial.jdbc.MyQueries"
42       classpathref="CLASSPATH" fork="true" dir=".">
43   <arg value="{PROPERTIESFILE}"/>
44 </java>
45 </target>
46

```

3. Modificando a consulta:

- Após analisar a solicitação, identificamos a ausência das tabelas **COFFEES** e **SUPPLIERS** em nosso banco de dados. Para atender à demanda, criamos essas tabelas conforme as especificações fornecidas na imagem disponível, utilizando um script SQL adequado.

```

1  -- Script para criar tabela SUPPLIERS
2  CREATE TABLE SUPPLIERS (
3      ID SERIAL PRIMARY KEY,
4      SUP_ID INTEGER UNIQUE,
5      SUP_NAME VARCHAR(100),
6      STREET VARCHAR(100),
7      CITY VARCHAR(100)
8  );
9
10 -- Script para criar tabela COFFEES
11 CREATE TABLE COFFEES (
12     ID SERIAL PRIMARY KEY,
13     COF_NAME VARCHAR(100),
14     SUP_ID INTEGER,
15     PRICE DOUBLE PRECISION,
16     SALES INT,
17     TOTAL DOUBLE PRECISION,
18     FOREIGN KEY (SUP_ID) REFERENCES SUPPLIERS (SUP_ID)
19 );

```

BANCO DE DADOS 2

T04 - Modificando o tutorial JDBC

- Após isso, executado script para popular o banco com as mesmas informações da imagem:

```
-- Script para inserir dados na tabela SUPPLIERS
INSERT INTO SUPPLIERS (SUP_ID, SUP_NAME, STREET, CITY) VALUES
(49, 'Superior Coffee', '1 Party Place', 'Mendocino'),
(101, 'Acme, Inc.', '99 Maraket Street', 'Groundsville'),
(150, 'The High Ground', '100 Coffee Lane', 'Meadows'),
(456, 'Restaurant Supplies, Inc', '200 Magnolia Street', 'Meadows'),
(927, 'Professional Kitchen', '300 Daisy Avenue', 'Groundsville');

-- Script para inserir dados na tabela COFFEES
INSERT INTO COFFEES (COF_NAME, SUP_ID, PRICE, SALES, TOTAL) VALUES
('Colombian', 101, 7.99, 0, 0),
('French_Roast', 49, 8.99, 0, 0),
('Espresso', 150, 9.99, 0, 0),
('Colombian_Decaf', 101, 8.99, 0, 0),
('French_Roast_Decaf', 49, 9.99, 0, 0);
```

- E enfim, realizado o seguinte script para listar o nome de todos os fornecedores de café acompanhado da quantidade de tipos de cafés vendidos para a loja:

```
-- Fornecedor de café acompanhados da quantidade de tipos de cafés vendidos para a loja

SELECT s.SUP_NAME AS Nome_Fornecedor, COUNT(c.COF_NAME) AS Quantidade_Tipos_Cafe
FROM SUPPLIERS s
JOIN COFFEES c ON s.SUP_ID = c.SUP_ID
GROUP BY s.SUP_NAME;
```

- Feito a alteração da consulta no arquivo **MyQueries.java** e modificado a saída para corresponder à consulta (comentários da classe somente no print):

```
public static void getMyData(Connection con) throws SQLException {
    Statement stmt = null;
    String query =
        "SELECT SUPPLIERS.SUP_NAME, COUNT(COFFEES.COF_NAME) AS Quantidade_Tipos_Cafe " +
        "FROM SUPPLIERS " +
        "LEFT JOIN COFFEES ON SUPPLIERS.SUP_ID = COFFEES.SUP_ID " +
        "GROUP BY SUPPLIERS.SUP_NAME"; // feito mudança da consulta

    try { // realizado mudança para exibir conforme consulta
        stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery(query);
        System.out.println("Fornecedores de café e quantidade de tipos de café vendidos: ");
        while (rs.next()) { // verifica se há mais linhas disponíveis no ResultSet para serem lidas.
            String supplierName = rs.getString("SUP_NAME"); // armazena o nome do Fornecedor
            int coffeeCount = rs.getInt("Quantidade_Tipos_Cafe"); // armazena a quantidade de tipos de café
            System.out.println("Fornecedor: " + supplierName + ", Quantidade de Tipos de Café: " + coffeeCount);
            // imprime o nome do fornecedor e a quantidade até acabar, e o loop finalizar
        }
    } catch (SQLException e) {
        JDBCUtilities.printSQLException(e);
    } finally {
        if (stmt != null) { stmt.close(); }
    }
}
```

BANCO DE DADOS 2

T04 - Modificando o tutorial JDBC

4. Compilado, reiniciado e executado o novo código:

```
railson@railson-virtual-machine:~/mydir/JDBCTutorial$ ant jar
Buildfile: /home/railson/mydir/JDBCTutorial/build.xml

compile:
[echo] JDBC Tutorial is using /usr/local/jdk/bin/javac as Java compiler

jar:
[jar] Building jar: /home/railson/mydir/JDBCTutorial/lib/JDBCTutorial.jar

BUILD SUCCESSFUL
Total time: 0 seconds
```

```
railson@railson-virtual-machine:~/mydir/JDBCTutorial$ ant setup
Buildfile: /home/railson/mydir/JDBCTutorial/build.xml

setup:

drop-tables:
[sql] Executing resource: /home/railson/mydir/JDBCTutorial/sql/javadb/drop
-tables.sql
[sql] 8 of 8 SQL statements executed successfully

build-tables:
[sql] Executing resource: /home/railson/mydir/JDBCTutorial/sql/javadb/crea
te-tables.sql
[sql] 8 of 8 SQL statements executed successfully

populate-tables:
[sql] Executing resource: /home/railson/mydir/JDBCTutorial/sql/javadb/popu
late-tables.sql
[sql] 40 of 40 SQL statements executed successfully

BUILD SUCCESSFUL
Total time: 1 second
```

```
railson@railson-virtual-machine:~/mydir/JDBCTutorial$ ant mycode
Buildfile: /home/railson/mydir/JDBCTutorial/build.xml

mycode:
[java] Set the following properties:
[java] dbms: derby
[java] driver: org.apache.derby.jdbc.EmbeddedDriver
[java] dbName: testdb
[java] useName:
[java] serverName: localhost
[java] portNumber: 3306
[java] Connected to database
[java] Fornecedores de café e quantidade de tipos de café vendidos:
[java] Fornecedor: Acme, Inc., Quantidade de Tipos de Café: 2
[java] Fornecedor: Professional Kitchen, Quantidade de Tipos de Café: 0
[java] Fornecedor: Restaurant Supplies, Inc., Quantidade de Tipos de Café:
0
[java] Fornecedor: Superior Coffee, Quantidade de Tipos de Café: 2
[java] Fornecedor: The High Ground, Quantidade de Tipos de Café: 1
[java] Releasing all open resources ...

BUILD SUCCESSFUL
Total time: 1 second
```

5. Download arquivo comp:

Feito o Download do arquivo e feito a modificação dos caminhos conforme meus repositórios:

```
Abrir  [icon] comp /home/railson/mydir/JDBCTutorial Salvar
1 #!/bin/sh -
2
3 derby=/usr/share/java/derby.jar
4 mypath=/home/railson/mydir/JDBCTutorial
5 mypackage=com/oracle/tutorial/jdbc
```

BANCO DE DADOS 2

T04 - Modificando o tutorial JDBC

6. Mudança de permissão e execução:

Foi feita a alteração da permissão do arquivo com o comando **chmod 755 comp** e após isso foi executado o comando **./comp MyQueries properties/javadb-sample-properties.xml** para rodar o script **comp** e fazer execução da classe **MyQueries**.

```
railson@railson-virtual-machine:~/mydir/JDBCTutorial$ chmod 755 comp
railson@railson-virtual-machine:~/mydir/JDBCTutorial$ ./comp MyQueries propertie
s/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
Fornecedores de café e quantidade de tipos de café vendidos:
Fornecedor: Acme, Inc., Quantidade de Tipos de Café: 2
Fornecedor: Professional Kitchen, Quantidade de Tipos de Café: 0
Fornecedor: Restaurant Supplies, Inc., Quantidade de Tipos de Café: 0
Fornecedor: Superior Coffee, Quantidade de Tipos de Café: 2
Fornecedor: The High Ground, Quantidade de Tipos de Café: 1
Releasing all open resources ...
railson@railson-virtual-machine:~/mydir/JDBCTutorial$
```