

Trabalho 4

Integrantes:

Maycon Douglas Batista dos Santos 11921BSI209

&

Matheus Costa Monteiro 12111BSI281.

Processando comando sql em JDBC

Passo 1. Baixar o código fonte da classe MyQueries.java e JDBCUtilities.java para o diretório raiz do tutorial. Em seguida posicione o cursor na pasta raiz do tutorial com o comando cd e digite:

- 1.cp MyQueries.java src/com/oracle/tutorial/jdbc/
- 2.cp JDBCUtilities.java src/com/oracle/tutorial/jdbc/

Resultado após a execução só comando:

```
maycon@maycon-Inspiron-3584: ~/5Periodo/BD2/JDBCTutorial
maycon@maycon-Inspiron-3584:~/5Periodo/BD2/JDBCTutorial$ ls src/com/oracle/tutorial/jdbc/
CachedRowSetSample.java  CoffeesTableModel.java  JDBCUtilities.java  RSSFeedsTable.java  WebRowSetSample.java
CityFilter.java         DatalinkSample.java     JDBCUtilities.java  StateFilter.java
ClobSample.java         ExampleRowSetListener.java  JoinSample.java    StoredProcedureJavaDBSample.java
CoffeesFrame.java       FilteredRowSetSample.java  MyQueries.java     StoredProcedureMySQLSample.java
CoffeesTable.java       JdbcRowSetSample.java     ProductInformationTable.java  SuppliersTable.java
```

Passo 2. Incluir uma sessão de configuração para executar a classe MyQueries no arquivo build.xml:

```
251
252      <!--Incluido na tarefa 4-->
253      <target name="mycode">
254          <java classname="com.oracle.tutorial.jdbc.MyQueries"
255              classpathref="CLASSPATH" fork="true" dir=".">
256              <arg value="${PROPERTIESFILE}"/>
257          </java>
258      </target>
```

Passo 3: Modificar a consulta presente em MyQueries.java para atender ao seguinte enunciado:Liste os nomes de todos os fornecedores de café acompanhados da quantidade de tipos de cafés vendidos para a loja. Abaixo estão as listagens das duas tabelas envolvidas:Listagem da tabela COFFEES:

Para resolução foi selecionado o nome do fornecedor do café que e um contador nos tipos de café retornados da tabela café que a chave estrangeira seja igual a da tabela fornecedores e o nome do fornecedor seja igual ao nome passado como parâmetro.

Consulta: `select s.SUP_NAME as SNAME, COUNT(c.COF_NAME) as CNAME from COFFEES as c, SUPPLIERS as s where s.SUP_ID = c.SUP_ID and s.SUP_NAME like '%Superior Coffee%' group by s.SUP_NAME;`

```
1  select s.SUP_NAME as SNAME, COUNT(c.COF_NAME) as CNAME
2  |   from COFFEES as c, SUPPLIERS as s
3  |   where s.SUP_ID = c.SUP_ID and s.SUP_NAME like '%Superior Coffee%'
4  |   group by s.SUP_NAME;
```

Passo 4: Compilar, reiniciar o banco e executar o novo código.

1. ant jar; ant setup; ant mycode

Execução do comando ant jar:

```
maycon@maycon-Inspiron-3584:~/5Periodo/BD2/JDBCTutorial$ ant jar
Buildfile: /home/maycon/5Periodo/BD2/JDBCTutorial/build.xml

compile:
[echo] JDBC Tutorial is using /usr/lib/jvm/java-17-openjdk-amd64/bin/javac as Java compiler
[javac] Compiling 1 source file to /home/maycon/5Periodo/BD2/JDBCTutorial/classes

jar:
[jar] Building jar: /home/maycon/5Periodo/BD2/JDBCTutorial/lib/JDBCTutorial.jar

BUILD SUCCESSFUL
Total time: 4 seconds
maycon@maycon-Inspiron-3584:~/5Periodo/BD2/JDBCTutorial$
```

Execução do comando ant setup:

```
maycon@maycon-Inspiron-3584:~/5Periodo/BD2/JDBCTutorial$ ant setup
Buildfile: /home/maycon/5Periodo/BD2/JDBCTutorial/build.xml

setup:

drop-tables:
[sql] Executing resource: /home/maycon/5Periodo/BD2/JDBCTutorial/sql/javadb/drop-tables.sql
[sql] 8 of 8 SQL statements executed successfully

build-tables:
[sql] Executing resource: /home/maycon/5Periodo/BD2/JDBCTutorial/sql/javadb/create-tables.sql
[sql] 8 of 8 SQL statements executed successfully

populate-tables:
[sql] Executing resource: /home/maycon/5Periodo/BD2/JDBCTutorial/sql/javadb/populate-tables.sql
[sql] 40 of 40 SQL statements executed successfully

BUILD SUCCESSFUL
Total time: 11 seconds
maycon@maycon-Inspiron-3584:~/5Periodo/BD2/JDBCTutorial$
```

Execução do comando ant mycode:

```
maycon@maycon-Inspiron-3584:~/5Periodo/BD2/JDBCTutorial$ ant mycode
Buildfile: /home/maycon/5Periodo/BD2/JDBCTutorial/build.xml

mycode:
[java] Set the following properties:
[java] dbms: derby
[java] driver: org.apache.derby.jdbc.EmbeddedDriver
[java] dbName: testdb
[java] userName:
[java] serverName: localhost
[java] portNumber: 3306
[java] Connected to database
[java] Coffees bought from Superior Coffee:
[java]     Superior Coffee count: 2
[java] Releasing all open resources ...

BUILD SUCCESSFUL
Total time: 4 seconds
maycon@maycon-Inspiron-3584:~/5Periodo/BD2/JDBCTutorial$
```

Passo 5. Baixar o arquivo comp para a pasta raiz do tutorial (mesma pasta onde está o build.xml). De agora em diante utilizaremos o comp para compilar e executar as modificações da classe MyQueries e evitar reutilizar o código armazenado na biblioteca JDBCTutorial.jar. Abram o script comp para entender como ele funciona. Lembre-se de alterar os caminhos absolutos que estão digitados no arquivo comp, esse arquivo foi configurado para a minha máquina e você deve alterar as pastas para a sua máquina.

Resultado:

```
maycon@maycon-Inspiron-3584:~/5Periodo/BD2/JDBCTutorial$ ls
build.xml  classes  comp  derby.log  lib  properties  sql  src  testdb  txt  xml
maycon@maycon-Inspiron-3584:~/5Periodo/BD2/JDBCTutorial$
```

Passo 6. Antes de executar o script comp, mude a permissão de execução no linux:

1. `chmod 755 comp`
2. `./comp MyQueries properties/javadb-sample-properties.xml`

Como vimos na figura do passo anterior o 'comp' já está com permissão de execução ilustrado pela cor verde.

Resultado após uso do segundo comando:

```
maycon@maycon-Inspiron-3584:~/5Periodo/BD2/JDBCTutorial$ ./comp MyQueries 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
Coffees bought from Superior Coffee:
    Superior Coffee count: 2
Releasing all open resources ...
maycon@maycon-Inspiron-3584:~/5Periodo/BD2/JDBCTutorial$
```