

Universidade Federal de Uberlândia

Bacharelado em sistemas de informação

Faculdade de computação

Criando e populando tabelas via JDBC

Uberlândia – MG

Universidade Federal de Uberlândia

Bacharelado em sistemas de informação

Faculdade de computação

Maycon douglasBatista dos santos11921BSI209

&

Matheus Costa Monteiro 12111BSI281.

Uberlândia – MG

Relatório

Documento de relatório para a preparação de ambiente de desenvolvimento, Trabalho 07 da disciplina GSI021 – Banco de Dados II do curso de Bacharelado em Sistemas de Informação.

Professor: Anderson Rodrigues Santos

Sumário

1.....	
2.....	
3.....	
4.....	
5.....	
6.....	
1.....	
2.....	
7.....	
1.....	
2.....	
3.....	
8.....	
9.....	

Objetivo

O objetivo deste trabalho é criarmos a partir de um arquivo com dados de criarmos uma nova tabela usando interface do pgadmin ou qualquer outro software, estabelecer uma conexão com esse banco e popularmos usando um arquivos de dados via JDBC.

Questão 1) Nesta atividade vamos utilizar uma nova versão do banco da Instituição Bancária (IB). Não adianta tentar inserir estes dados no banco das aulas anteriores, visto que acontecerão erros de integridade referencial para todas as linhas da nova tabela que vamos criar e popular. Portanto, baixe o arquivo IB2.dump. Na sequência você criará um banco vazio para importaro dump para o novo BD de nome IB2:

1. <https://www.dropbox.com/s/6zuffauqzvd4fde/IB2.dump>

```
maycon@maycon-Inspiron-3584:~/Downloads$ cd ../5periodo/BD2/
maycon@maycon-Inspiron-3584:~/5periodo/BD2$ ls
gsi021_-_banco_de_dados_2.pdf  IB.dump  Listas  postgresql-14-A4.pdf  Tarefa2  Tarefa4
IB2.dump                      JDBCTutorial  material  Tarefa1  Tarefa3  Tarefa5
maycon@maycon-Inspiron-3584:~/5periodo/BD2$
```

Questão 2)

Criar um banco de dados vazio na interface do pgadmin3 de nome IB2.

```
postgres=# create database IB2;
CREATE DATABASE
postgres=# \l
```

List of databases							
Name	Owner	Encoding	Collate	Ctype	ICU Locale	Locale Provider	Access privileges
IB	postgres	UTF8	pt_BR.UTF-8	pt_BR.UTF-8		libc	
ib2	postgres	UTF8	pt_BR.UTF-8	pt_BR.UTF-8		libc	
postgres	postgres	UTF8	pt_BR.UTF-8	pt_BR.UTF-8		libc	
template0	postgres	UTF8	pt_BR.UTF-8	pt_BR.UTF-8		libc	=c/postgres + postgres=CTc/postgres
template1	postgres	UTF8	pt_BR.UTF-8	pt_BR.UTF-8		libc	=c/postgres + postgres=CTc/postgres

(5 rows)

```
postgres=#
```

Questão 3)

Executar a carga do arquivo IB2.dump no banco IB2;

1. `psql -h localhost -p 5432 -U postgres IB2 -f IB2.dump`

```

maycon@maycon-Inspiron-3584:~/5periodo/BD2$ psql -h localhost -p 5432 -U postgres ib2 -f IB2.dump
Password for user postgres:
SET
SET
SET
SET
SET
CREATE EXTENSION
COMMENT
SET
CREATE FUNCTION
ALTER FUNCTION
CREATE FUNCTION
ALTER FUNCTION
CREATE FUNCTION
ALTER FUNCTION
CREATE FUNCTION
ALTER FUNCTION
CREATE FUNCTION
ALTER FUNCTION
SET
SET
CREATE TABLE
ALTER TABLE
CREATE TABLE
ALTER TABLE
CREATE TABLE
ALTER TABLE
CREATE TABLE
ALTER TABLE
CREATE TABLE
ALTER TABLE
CREATE SEQUENCE
ALTER TABLE

```

4) Baixar o arquivo debito-populate-table.txt do link abaixo para a pasta raiz do tutorial do JDBC:

1. <https://www.dropbox.com/s/3nommkws4zqgvvgq/debito-populate-table.txt>

```

maycon@maycon-Inspiron-3584:~/5periodo/BD2$ cp ~/Downloads/debito-populate-table.txt .
maycon@maycon-Inspiron-3584:~/5periodo/BD2$ ls
debito-populate-table.txt  IB2.dump  JDBCTutorial  material  Tarefa1  Tarefa3  Tarefa5
gsi021_-_banco_de_dados_2.pdf  IB.dump  Listas  postgresql-14-A4.pdf  Tarefa2  Tarefa4  Tarefa7
maycon@maycon-Inspiron-3584:~/5periodo/BD2$ mv debito-populate-table.txt JDBCtutorial/
maycon@maycon-Inspiron-3584:~/5periodo/BD2$ ls JDBCtutorial/
BD2-14-postgresql-42.2.4.jar  classes  debito-populate-table.txt  JDBCtutorial.iml  properties  src  txt
build.xml  comp  derby.log  lib  sql  testdb  xml
maycon@maycon-Inspiron-3584:~/5periodo/BD2$

```

2. Listar o conteúdo do arquivo baixado com cat ou editando-o com gedit;

```

maycon@maycon-Inspiron-3584:~/5periodo/BD2$ cat ~/Downloads/debito-populate-table.txt
1001 900.75 4 2011-12-24 759 Pampulha Pedro Alvares Sousa
1002 858.81 5 2012-11-18 34526 Central Marcos Andrade
1003 353.28 5 2011-9-12 62195 Pampulha Reinaldo Pereira da Silva
1004 1422.57 4 2012-7-15 45543 Gameleira Elvis Fernando Da Silva
1005 1302.97 5 2012-3-2 22201 Central Joaquim Carlos Reis
1006 268.32 3 2013-11-21 2370 Central Geraldo Oliveira
1007 1199.2 4 2010-10-8 42436 Central Andrade de Freitas
1008 963.43 1 2013-9-29 21707 Gameleira Bruno Tadeu Pita
1009 1114.45 5 2011-11-23 46539 Central Alexandre Márcio de Souza
1010 647.93 4 2011-7-18 68348 Cidade Jardim Rafael Mesquita Gondim
1011 322.39 2 2011-4-13 57745 Cidade Jardim Maria Raquel Teixeira de Castro
1012 354.2 3 2012-8-7 55878 Pampulha Thiago Leôncio Guimarães
1013 219.93 2 2011-9-27 70612 Pampulha Rafael Mesquita Gondim
1014 1081.85 5 2012-11-7 1767 Gameleira Manuel Oliveira
1015 571.7 5 2013-4-5 76313 UFU Bruno Miranda Pacheco de Castro
1016 572.33 2 2011-1-25 55207 Cidade Jardim Marta Avelar Santos
1017 1462.73 3 2012-8-19 69118 Central João Boiadeiro
1018 201.53 2 2011-8-21 72069 Central Andrade de Freitas
1019 376.18 3 2011-6-14 20195 Cidade Jardim Norton Saint Clair Silva
1020 1238.17 5 2010-7-4 74545 Pampulha Maria Raquel Teixeira de Castro
1021 1320.77 5 2011-8-15 46248 UFU Carla Soares Sousa
1022 804.21 1 2010-11-2 99737 Cidade Jardim Pedro Alvares Sousa
1023 677.68 4 2011-4-6 51643 Glória Felipe Peres Bichara Júnior
1024 145.83 2 2011-5-21 8072 Gameleira Germano Luiz de Paula
1025 1251.78 2 2011-6-21 8275 Glória Marco Aurélio Santos
1026 684.66 4 2013-9-19 17477 Glória Gustavo Baer Albuquerque
1027 1227.99 5 2012-8-13 93125 Gameleira André Cabral da Silva
1028 846.33 4 2011-2-25 8275 Glória Marco Aurélio Santos
1029 448.13 5 2011-7-24 74545 Pampulha Maria Raquel Teixeira de Castro
1030 1401.8 2 2013-5-20 50073 Cidade Jardim André Cabral da Silva
1031 1454.12 6 2012-1-8 30273 Glória Jefferson Oliveira
1032 584.67 5 2011-5-1 3682 UFU Maria Lúcia Alves
1033 1298.39 4 2012-9-21 30273 Glória Jefferson Oliveira
1034 553.1 4 2010-11-14 34787 UFU Joaquim Carlos Reis
1035 263 3 2012-2-19 759 Pampulha Pedro Alvares Sousa

```

Questão 5)

Crie um arquivo vazio com o gedit, insira o seguinte código no arquivo vazio e salve-o com o nome de ReadFile.java, na mesma pasta na qual o arquivo debito-populate-table.txt foi salvo:

```
BD2 > JDBCTutorial > J ReadFile.java > ReadFile > main(String[])
1  import java.io.FileReader;
2  import java.io.BufferedReader;
3  import java.io.IOException;
4  import java.util.Scanner;
5
6  public class ReadFile {
7      public static void main(String[] args) throws IOException {
8          BufferedReader inputStream = null;
9          Scanner scanned_line = null;
10         String line;
11         String[] value;
12         value = new String[7];
13
14         int countv;
15         try {
16             inputStream = new BufferedReader(new FileReader("debito-populate-table.txt"));
17             while ((line = inputStream.readLine()) != null) {
18                 countv=0;
19                 System.out.println("<<");
20                 //split fields separated by tab delimiters
21                 scanned_line = new Scanner(line);
22                 scanned_line.useDelimiter("\t");
23                 while (scanned_line.hasNext()) {
24                     System.out.println(value[countv++]=scanned_line.next());
25                 } //while
26
27                 if (scanned_line != null) { scanned_line.close();}
28                 System.out.println(">>");
29                 System.out.println("insert into debito (numero_debito, valor_debito, motivo_debito, data_debito, numero_conta, nome_agencia, nome_cliente)
30                     + "values (" + value[0] +", " + value[1] +", " + value[2] +", " + value[3] +", " + value[4] +", " + value[5] +", " + value[6] + "));"
31             } //while
32         } finally { if (inputStream != null) { inputStream.close(); } } //if & finally
33     } //main
34 } //class
```

Questão 6) Compile e execute o arquivo ReadFile.java:

Passo 1. javac ReadFile.java

Passo 2. java ReadFile

```
maycon@maycon-Inspiron-3584:~/5periodo/BD2/JDBCTutorial$ javac ReadFile.java
maycon@maycon-Inspiron-3584:~/5periodo/BD2/JDBCTutorial$ java ReadFile
<<
1001
900.75
4
2011-12-24
759
Pampulha
Pedro Alvares Sousa
>>
insert into debito (numero_debito, valor_debito, motivo_debito, data_debito, numero_conta, nome_agencia, nome_cliente) values (1001, 900.75, 4
, '2011-12-24', 759, 'Pampulha', 'Pedro Alvares Sousa');
<<
1002
858.81
5
2012-11-18
34526
Central
Marcos Andrade
>>
insert into debito (numero_debito, valor_debito, motivo_debito, data_debito, numero_conta, nome_agencia, nome_cliente) values (1002, 858.81, 5
, '2012-11-18', 34526, 'Central', 'Marcos Andrade');
<<
1003
353.28
5
2011-9-12
62195
Pampulha
Reinaldo Pereira da Silva
>>
insert into debito (numero_debito, valor_debito, motivo_debito, data_debito, numero_conta, nome_agencia, nome_cliente) values (1003, 353.28, 5
, '2011-9-12', 62195, 'Pampulha', 'Reinaldo Pereira da Silva');
<<
```


Questão 7)

7) Utilize a interface do *pgadmin3* para criar a seguinte tabela no banco:

Nome da tabela: debito

Nome da Coluna	Tipo	Modificadores
numero_debito	integer	not null
valor_debito	double precision	not null
motivo_debito	smallint	
data_debito	date	
numero_conta	integer	
nome_agencia	character varying(50)	
nome_cliente	character varying(80)	

Índices:

Nome: pk_debito, **Tipo:** PRIMARY KEY, **Sobre:** (numero_debito)

Restrições de chave estrangeira:

Nome: fk_debito, **Tipo:** FOREIGN KEY **Sobre:** (numero_conta, nome_agencia, nome_cliente), **Referenciando:** conta(numero_conta, nome_agencia, nome_cliente)

Perceba que ao final temos apenas uma tabela sem dado algum e será necessário realizar um carga de dados na tabela.

```
create table debito
(
    numero_debito integer not null,
    valor_debito double precision not null,
    motivo_debito smallint,
    data_debito date,
    numero_conta integer,
    nome_agencia character varying(50),
    nome_cliente character varying(80),
    primary key (numero_debito),
    foreign key (nome_agencia, numero_conta, nome_cliente) references conta(nome_agencia, numero_conta, nome_cliente)
);
```

Questão 8)

Após ter criado manualmente a tabela, agora é necessário populá-la com os dados do arquivo debito-populate.txt. Crie um método na classe MyQueries para executar tal operação. Utilize o arcabouço do código abaixo para este propósito.

```

public static void populateTable (Connection con) throws SQLException{
    Statement stmt = null;
    String query = "";

    try{
        stmt = con.createStatement();
        System.out.println("Executando DDL/DML");
        stmt.executeUpdate(query);
    } catch (SQLException e){
        JDBCUtilities.printSQLException(e);
    } finally{
        if(stmt != null) {stmt.close();}
    }
}

```

Questão 9) O código ReadFile.java deve ser reaproveitado para popular a tabela 'debito'. Para tanto:

```

48 public static List<String> readTXT () throws IOException {
49     BufferedReader inputStream = null;
50     Scanner scanned_line = null;
51     String line;
52     String[] value;
53     value = new String[7];
54
55     int countv;
56     List<String> lista = new ArrayList<>();
57     try {
58         inputStream = new BufferedReader(new FileReader("debito-populate-table.txt"));
59         while ((line = inputStream.readLine()) != null) {
60             countv=0;
61
62             // System.out.println("<<");
63             // split fields separated by tab delimiters
64             scanned_line = new Scanner(line);
65             scanned_line.useDelimiter("\t");
66             while (scanned_line.hasNext()) {
67                 value[countv++]=scanned_line.next();
68             } //while
69
70             if (scanned_line != null) { scanned_line.close();}
71             //System.out.println(">>");
72             lista.add("insert into debito (numero_debito, valor_debito, motivo_debito, data_debito, numero_conta, nome_agencia, nome_cliente) "
73                 + "values (" + value[0] + ", " + value[1] + ", " + value[2] + ", " + value[3] + ", " + value[4] + ", " + value[5] + ", " + value[6] + ");");
74         } //while
75     } finally { if (inputStream != null) {
76         inputStream.close();
77     }} //if & finally
78
79     return lista;
80 } //main
81
82 public static void populateTable (Connection con) throws SQLException{

```

O arquivo ReadFile foi colocado no mesmo caminho que o MyQueries2 já o arquivo que servirá para popular a tabela foi deixado na pasta raiz do tutorial, e o código foi alterado para que seja retornado uma lista com as string de inserções.

Agora nosso método irá receber uma lista com as string contendo o comando de inserção que rodará um por um.

```

82
83     public static void populateTable (Connection con) throws SQLException{
84         Statement stmt = null;
85
86         try{
87             stmt = con.createStatement();
88             System.out.println("Executando DDL/DML");
89             List <String> querys = readTXT();
90
91             if(querys.size() > 0){
92                 for(String item : querys){
93                     stmt.executeUpdate(item);
94                 }
95             }
96         }catch(SQLException e){
97             JDBCUtilities.printSQLException(e);
98         }catch(IOException e){
99             e.printStackTrace();
100         }finally{
101             if(stmt != null) {stmt.close();}
102         }
103     }

```

Questão 10)

Modificar/criar o arquivo de propriedades de conexão para acessar o banco IB2.

```

BD2 > JDBCTutorial > properties > 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/maycon/5Periodo/BD2/JDBCTutorial/BD2-14-postgresql-42.2.4.jar</entry>
7      <entry key="driver">org.postgresql.Driver</entry>
8      <entry key="database_name">ib2</entry>
9      <entry key="user_name">postgres</entry>
10     <entry key="password">pgsql1312531514</entry>
11     <entry key="server_name">localhost</entry>
12     <entry key="port_number">5432</entry>
13
14 </properties>
15

```

Veja que somente foi mudado o nome do bando de IB para ib2.

Questão 11)

Salve, compile e execute o MyQueries.

1. Certifique-se de que o arquivo properties/postgres-properties.xml está configurado com os parâmetros de conexão ao banco de dados correto;
2. ./comp MyQueries properties/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/maycon/5Periodo/BD2/JDBCTutorial/BD2-14-postgresql-42.2.4.jar</entry>
7      <entry key="driver">org.postgresql.Driver</entry>
8      <entry key="database_name">ib2</entry>
9      <entry key="user_name">postgres</entry>
10     <entry key="password">pgsql1312531514</entry>
11     <entry key="server_name">localhost</entry>
12     <entry key="port_number">5432</entry>
13
14 </properties>
15

```

Arquivo devidamente configurado.

```

maycon@maycon-Inspiron-3584:~/5periodo/BD2/JDBCTutorial$ ./comp MyQueries2 properties/postgres-properties.xml
Set the following properties:
dbms: postgresql
driver: org.postgresql.Driver
dbName: ib2
userName: postgres
serverName: localhost
portNumber: 5432
Connected to database
Executando DDL/DML
Releasing all open resources ...
maycon@maycon-Inspiron-3584:~/5periodo/BD2/JDBCTutorial$

```

Execução do comando `./comp MyQueries2 properties/postgres-properties.xml` feita com sucesso.

12)

Sempre que você tentar executar a inserção destes dados mais de uma vez na tabela debito serão exibidas mensagens de erro como esta:

org.postgresql.util.PSQLException: ERROR: duplicate key value violates unique constraint "pk_debito"

Execute o código e comprove.

`./comp MyQueries2 properties/postgres-properties.xml`

Resultado:

```

maycon@maycon-Inspiron-3584:~/Speriodo/BD2/JDBCTutorial$ ./comp MyQueries2 properties/postgres-properties.xml
Set the following properties:
dbms: postgresql
driver: org.postgresql.Driver
dbName: ib2
userName: postgres
serverName: localhost
portNumber: 5432
Connected to database
Executando DDL/DML
org.postgresql.util.PSQLException: ERROR: duplicate key value violates unique constraint "debito_pkey"
  Detalhe: Key (numero_debito)=(1001) already exists.
    at org.postgresql.core.v3.QueryExecutorImpl.receiveErrorResponse(QueryExecutorImpl.java:2440)
    at org.postgresql.core.v3.QueryExecutorImpl.processResults(QueryExecutorImpl.java:2183)
    at org.postgresql.core.v3.QueryExecutorImpl.execute(QueryExecutorImpl.java:308)
    at org.postgresql.jdbc.PgStatement.executeInternal(PgStatement.java:441)
    at org.postgresql.jdbc.PgStatement.execute(PgStatement.java:365)
    at org.postgresql.jdbc.PgStatement.executeWithFlags(PgStatement.java:307)
    at org.postgresql.jdbc.PgStatement.executeCachedSql(PgStatement.java:293)
    at org.postgresql.jdbc.PgStatement.executeWithFlags(PgStatement.java:270)
    at org.postgresql.jdbc.PgStatement.executeUpdate(PgStatement.java:244)
    at com.oracle.tutorial.jdbc.MyQueries2.populateTable(MyQueries2.java:93)
    at com.oracle.tutorial.jdbc.MyQueries2.main(MyQueries2.java:126)
SQLState: 23505
Error Code: 0
Message: ERROR: duplicate key value violates unique constraint "debito_pkey"
  Detalhe: Key (numero_debito)=(1001) already exists.
Releasing all open resources ...

```

Isso ocorre porque a integridade da chave primária é policiada pelo SGBD. Para evitar que esta mensagem de erro ocorra, supondo que a tabela precisa ser recarregada periodicamente, podemos usar um comando para truncar a tabela antes de novas inserções. Insira o comando antes do loop de inserção de linhas lidas do arquivo, salve e execute novamente:

1. `stmt.executeUpdate("truncate table debito;");`

```

83     public static void populateTable (Connection con) throws SQLException{
84         Statement stmt = null;
85
86         try{
87             stmt = con.createStatement();
88             System.out.println("Executando DDL/DML");
89             stmt.executeUpdate("truncate table debito;");
90             List <String> querys = readTXT();
91
92             if(querys.size() > 0){
93                 for(String item : querys){
94                     stmt.executeUpdate(item);
95                 }
96             }
97         }catch(SQLException e){
98             JDBCUtilities.printSQLException(e);
99         }catch(IOException e){
100             e.printStackTrace();
101         }finally{
102             if(stmt != null) {stmt.close();}
103         }
104     }

```

2. `./comp MyQueries properties/postgres-properties.xml`

```
maycon@maycon-Inspiron-3584:~/5periodo/BD2/JDBCTutorial$ ./comp MyQueries2 properties/postgres-properties.xml
Set the following properties:
dbms: postgresql
driver: org.postgresql.Driver
dbName: ib2
userName: postgres
serverName: localhost
portNumber: 5432
Connected to database
Executando DDL/DML
Releasing all open resources ...
```

Todo o código completo pode ser encontrado no repositório no github no endereço:
<https://github.com/mayconbaptista/5periodo/tree/main/BD2>