

# BANCO DE DADOS 2

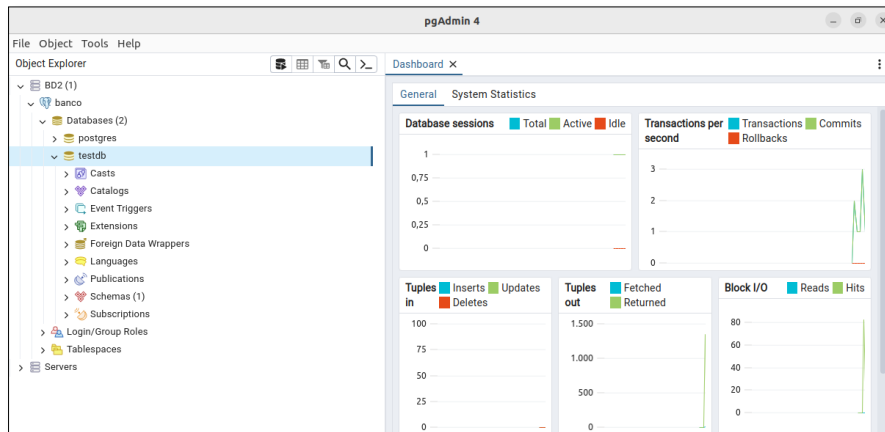
## T05 – Estabelecendo conexões

Railson Da Silva Martins - 11811BSI208

Wothon Mateus de Araujo - 12111BSI262

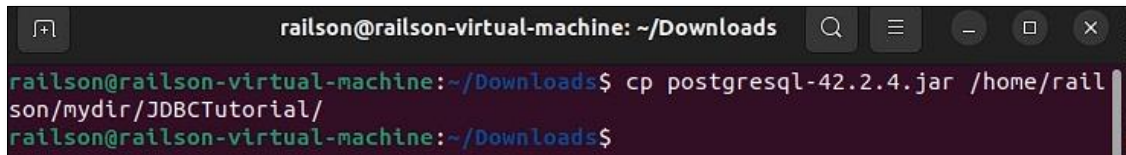
### 1. Conexão ao SGBD.

Realizada conexão com SGBD **pgadmin4** após criação do banco de dados **testedb** para o **Postgres** utilizando a porta 5432:



### 2. Download e cópia.

Realizado Download do arquivo **postgresql-42.2.4.jar**, localizado na pasta que foi baixado e copiado para pasta do **JDBCTutorial**.



### 3. Configurando propriedades Postgres.

Realizada configuração das propriedades para o PostgreSQL.



#### 4. Alterando JDBCUtilities.java.

Alterada estrutura para conexão com **PostgreSQL**:

```

157 String currentUrlString = null;
158
159 if (this.dbms.equals("mysql")) {
160     currentUrlString = "jdbc:" + this.dbms + "://" + this.serverName + ":" + this.portNumber
161 + "/" ;
162     conn = DriverManager.getConnection(currentUrlString, connectionProps);
163     this.urlString = currentUrlString + this.dbName;
164     conn.setCatalog(this.dbName);
165 } else if (this.dbms.equals("derby")) {
166     this.urlString = "jdbc:" + this.dbms + ":" + this.dbName;
167     conn = DriverManager.getConnection(this.urlString + ";create=true", connectionProps);
168 } else if (this.dbms.equals("postgresql")) {
169     currentUrlString = "jdbc:" + this.dbms + "://" + this.serverName + ":" + this.portNumber
170 + "/" + this.dbName;
171     conn = DriverManager.getConnection(currentUrlString, connectionProps);
172     this.urlString = currentUrlString + this.dbName;
173     conn.setCatalog(this.dbName);
174
175 System.out.println("Connected to database");
176 return conn;
177 }

```

#### 5. Copiando e modificando novas classes **MyQueries2.java** e **MyQueries3.java**:

- Realizado a cópia da classe com os comandos **cp MyQueries.java MyQueries2.java** e **cp MyQueries.java MyQueries3.java**. Modificados nomes das classes dentro dos arquivos também.

- Implementado na classe **MyQueries2.java** o retorno de **clientes que possuem apenas depósitos com sua respectiva soma** com a seguinte consulta e seus respectivos resultados:

```

SELECT c.nome_cliente, SUM(d.saldo_deposito) AS soma_de_depositos
FROM cliente c
INNER JOIN deposito d ON c.nome_cliente = d.nome_cliente
LEFT JOIN emprestimo e ON c.nome_cliente = e.nome_cliente
WHERE e.numero_emprestimo IS NULL
GROUP BY c.nome_cliente;

```

|   | nome_cliente<br>[PK] character varying (80) | soma_de_depositos<br>double precision |
|---|---|---------------------------------------|
| 1 | Alexandre Marcio de Souza                   | 1201.37                               |
| 2 | Cláudia Santos Mota                         | 3599.29                               |
| 3 | Felippe Peres Bichara Junior                | 7417.87                               |
| 4 | Thiago Andrade Fiuza                        | 479.66                                |
| 5 | Wantuil Diniz e Souza                       | 84.17                                 |

## BANCO DE DADOS 2

### T05 – Estabelecendo conexões

Implementado da seguinte forma na classe **MyQueries2.java**:

```
public static void getMyData(Connection con) throws SQLException {
    // Declaração de um objeto Statement inicializado como nulo
    Statement stmt = null;
    // Definição da consulta SQL a ser executada
    String query =
        "SELECT c.nome_cliente, SUM(d.saldo_deposito) AS soma_de_depositos " +
        "FROM cliente c " +
        "INNER JOIN deposito d ON c.nome_cliente = d.nome_cliente " +
        "LEFT JOIN emprestimo e ON c.nome_cliente = e.nome_cliente " +
        "WHERE e.numero_emprestimo IS NULL " + // Filtro para garantir que o cliente não tenha empréstimos
        "GROUP BY c.nome_cliente;"; // Agrupamento por nome do cliente para calcular a soma dos depósitos de cada cliente

    try {
        // Inicialização do objeto Statement usando a conexão fornecida
        stmt = con.createStatement();
        // Execução da consulta SQL e armazenamento do resultado em um objeto ResultSet
        ResultSet rs = stmt.executeQuery(query);

        System.out.println("Clientes com depósitos e suas somas de depósitos:");
        while (rs.next()) { // Loop para rodar em toda linha da consulta
            // Extração do nome do cliente e da soma dos depósitos da linha atual do ResultSet
            String clienteNome = rs.getString("nome_cliente");
            double somaDepositos = rs.getDouble("soma_de_depositos");

            // Impressão do nome do cliente e a soma dos depósitos associados a ele
            System.out.println("Cliente: " + clienteNome + " - Soma de Depósitos: " + somaDepositos);
        }
    } catch (SQLException e) {
        JDBCUtilities.printSQLException(e);
    } finally {
        if (stmt != null) {
            stmt.close();
        }
    }
}
```

- Implementado na classe **MyQueries3.java** o retorno de **clientes que possuem depósitos e empréstimos com as respectivas somas** com a seguinte consulta e seus respectivos resultados:

```
WITH SomaDepositos AS (SELECT c.nome_cliente, COALESCE(SUM(d.saldo_deposito), 0) AS soma_de_depositos
FROM cliente c LEFT JOIN deposito d ON c.nome_cliente = d.nome_cliente
GROUP BY c.nome_cliente), SomaEmprestimos AS (SELECT c.nome_cliente,
COALESCE(SUM(e.valor_emprestimo), 0) AS soma_de_emprestimos
FROM cliente c LEFT JOIN emprestimo e ON c.nome_cliente = e.nome_cliente GROUP BY c.nome_cliente)
SELECT d.nome_cliente, d.soma_de_depositos, e.soma_de_emprestimos FROM SomaDepositos d
JOIN SomaEmprestimos e ON d.nome_cliente = e.nome_cliente
WHERE d.soma_de_depositos > 0 AND e.soma_de_emprestimos > 0;
```

|    | nome_cliente<br>[PK] character varying (80) | soma_de_depositos<br>double precision | soma_de_emprestimos<br>double precision |
|----|---|---------------------------------------|---|
| 1  | Thiago Leôncio Guimarães                    | 1753.42                               | 1411.99                                 |
| 2  | Everardo Monfort Leitão                     | 3562.54                               | 1886.98                                 |
| 3  | Carolina Soares                             | 3516.28                               | 1324.74                                 |
| 4  | Marcos Andrade                              | 3493.63                               | 886.85                                  |
| 5  | Germano Luiz de Paula                       | 104817.72                             | 1675.65                                 |
| 6  | Bruno Miranda Pacheco de Castro             | 2512.88                               | 1139.08                                 |
| 7  | Joaquim Carlos Reis                         | 498.56                                | 805.88                                  |
| 8  | Andre Cabral da Silva                       | 4490.47                               | 2612.98                                 |
| 9  | Marco Aurélio Santos                        | 3164.44                               | 3775.6399999999994                      |
| 10 | Gilmar Negreiros Carvalho                   | 5115.06                               | 2117.05                                 |
| 11 | Clayton Pereira Bonfim                      | 3231.84                               | 896.28                                  |

## BANCO DE DADOS 2

### T05 – Estabelecendo conexões

Implementado da seguinte forma na classe **MyQueries3.java**:

```
public static void getMyData(Connection con) throws SQLException {
    Statement stmt = null;
    String query =
        "WITH SomaDepositos AS (SELECT c.nome_cliente, COALESCE(SUM(d.saldo_deposito), 0) AS soma_de_depositos " +
        "FROM cliente c LEFT JOIN deposito d ON c.nome_cliente = d.nome_cliente " +
        "GROUP BY c.nome_cliente), " +
        "SomaEmprestimos AS (SELECT c.nome_cliente, COALESCE(SUM(e.valor_emprestimo), 0) AS soma_de_emprestimos " +
        "FROM cliente c LEFT JOIN emprestimo e ON c.nome_cliente = e.nome_cliente " +
        "GROUP BY c.nome_cliente) " + // Consulta principal que junta os resultados das CTEs
        "SELECT d.nome_cliente, d.soma_de_depositos, e.soma_de_emprestimos " +
        "FROM SomaDepositos d " +
        "JOIN SomaEmprestimos e ON d.nome_cliente = e.nome_cliente " +
        "WHERE d.soma_de_depositos > 0 AND e.soma_de_emprestimos > 0;" ;
    // Filtro para incluir apenas clientes com depósitos e empréstimos

    try {
        // Inicialização do objeto Statement usando a conexão fornecida
        stmt = con.createStatement();
        // Execução da consulta SQL e armazenamento do resultado em um objeto ResultSet
        ResultSet rs = stmt.executeQuery(query);
        System.out.println("Clientes com depósitos e empréstimos e suas somas:");
        // Iteração sobre o ResultSet para processar cada linha de resultado
        while (rs.next()) {
            // Extração do nome do cliente, soma de depósitos e soma de empréstimos da linha atual do ResultSet
            String clienteNome = rs.getString("nome_cliente");
            double somaDepositos = rs.getDouble("soma_de_depositos");
            double somaEmprestimos = rs.getDouble("soma_de_emprestimos");
            // Impressão do nome do cliente, soma de depósitos e soma de empréstimos associados a ele
            System.out.println("Cliente: " + clienteNome + " - Soma de Depósitos: " + somaDepositos +
                " - Soma de Empréstimos: " + somaEmprestimos);
        }
    } catch (SQLException e) { // Tratamento de exceção
        JDBCUtilities.printSQLException(e);
    } finally {
        if (stmt != null) {
            stmt.close();
        }
    }
}
```

## 6. Modificando e compilando arquivos:

- Alterado arquivo **comp** para utilização do **postgres** conforme informado e atribuído permissão de execução via comando **chmod 755 comp**.

```
1 #!/bin/sh -
2
3 derby=/usr/share/java/derby.jar
4 postgres=/home/railson/mydir/JDBCTutorial/postgresql-42.2.4.jar
5 mypath=/home/railson/mydir/JDBCTutorial
6 mypackage=com/oracle/tutorial/jdbc
7
8 if [ -r $mypath/src/$mypackage/$1.java -a -r $mypath/$2 ]
9 then
10
11     javac -cp "$postgres:$mypath/classes:$mypath/lib/JDBCTutorial.jar" $mypath/src/
    $mypackage/$1.java
12
13     if [ -r $mypath/src/$mypackage/$1.class ]
14     then
15         mv $mypath/src/$mypackage/$1.class $mypath/classes/$mypackage/
16
17         java -cp "$postgres:$mypath/classes" $mypackage/$1 $mypath/$2
18     fi
19 else
20     echo Missing file, check:
21     ls $mypath/src/$mypackage/$1.java
22     ls $mypath/$2
23 fi
24
```



## BANCO DE DADOS 2

### T05 – Estabelecendo conexões

- Compilado a classe **JDBCUtilities.java** com o comando **./comp JDBCUtilities properties/postgres-properties.xml**.

```
railson@railson-virtual-machine:~/mydir/JDBCTutorial$ ./comp JDBCUtilities properties/postgres-properties.xml
Reading properties file /home/railson/mydir/JDBCTutorial/properties/postgres-properties.xml
Set the following properties:
dbms: postgresql
driver: com.postgresql.cj.jdbc.Driver
dbName: testdb
userName: postgres
serverName: localhost
portNumber: 5432
Connected to database
Releasing all open resources ...
```

- Compilado a classe **MyQueries2.java** com o comando **./comp MyQueries2 properties/postgres-properties.xml**:

```
railson@railson-virtual-machine:~/mydir/JDBCTutorial$ ./comp MyQueries2 properties/postgres-properties.xml
Set the following properties:
dbms: postgresql
driver: com.postgresql.cj.jdbc.Driver
dbName: testdb
userName: postgres
serverName: localhost
portNumber: 5432
Connected to database
Clientes com depósitos e suas somas de depósitos:
Cliente: Alexandre Marcio de Souza - Soma de Depósitos: 1201.37
Cliente: Cláudia Santos Mota - Soma de Depósitos: 3599.29
Cliente: Felipe Peres Bichara Junior - Soma de Depósitos: 7417.87
Cliente: Thiago Andrade Fiuza - Soma de Depósitos: 479.66
Cliente: Wantuil Diniz e Souza - Soma de Depósitos: 84.17
Releasing all open resources ...
```

- Compilado a classe **MyQueries3.java** com o comando **./comp MyQueries3 properties/postgres-properties.xml**:

```
railson@railson-virtual-machine:~/mydir/JDBCTutorial$ ./comp MyQueries3 properties/postgres-properties.xml
Set the following properties:
dbms: postgresql
driver: com.postgresql.cj.jdbc.Driver
dbName: testdb
userName: postgres
serverName: localhost
portNumber: 5432
Connected to database
Clientes com depósitos e empréstimos e suas somas:
Cliente: Thiago Leôncio Guimarães - Soma de Depósitos: 1753.42 - Soma de Empréstimos: 1411.99
Cliente: Everardo Monfort Leitão - Soma de Depósitos: 3562.54 - Soma de Empréstimos: 1886.98
Cliente: Carolina Soares - Soma de Depósitos: 3516.28 - Soma de Empréstimos: 1324.74
Cliente: Marcos Andrade - Soma de Depósitos: 3493.63 - Soma de Empréstimos: 886.85
Cliente: Germano Luiz de Paula - Soma de Depósitos: 104817.72 - Soma de Empréstimos: 1675.65
Cliente: Bruno Miranda Pacheco de Castro - Soma de Depósitos: 2512.88 - Soma de Empréstimos: 1139.08
Cliente: Joaquim Carlos Reis - Soma de Depósitos: 498.56 - Soma de Empréstimos: 805.88
Cliente: Andre Cabral da Silva - Soma de Depósitos: 4490.47 - Soma de Empréstimos: 2612.98
Cliente: Marco Aurélio Santos - Soma de Depósitos: 3164.44 - Soma de Empréstimos: 3775.6399999999994
Cliente: Gilmar Negreiros Carvalho - Soma de Depósitos: 5115.06 - Soma de Empréstimos: 2117.05
Cliente: Clayton Pereira Bonfim - Soma de Depósitos: 3231.84 - Soma de Empréstimos: 896.28
Releasing all open resources ...
```

**Obs:** Para sucesso na compilação das classes, foi necessário mudança do nome do banco de IB para testdb e necessário inclusão do testdb.dump disponibilizado na aba geral do teams para atender as demandas do banco no postgres de todos os “MyQueries?.java”.