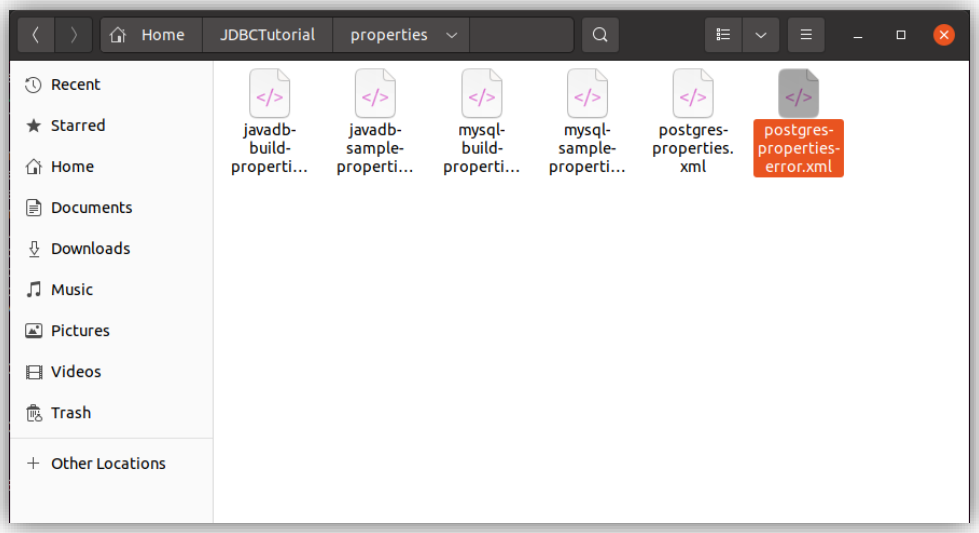
Sistemas de Bancos de Dados

Aula 09 – Tratamento de erros de conexão

Murielly Oliveira Nascimento – 11921BSI22

1) Copiar o arquivo postgres-properties.xml confeccionado na aula anterior para a pasta properties do tutorial JDBC sob o nome de postgres-properties-error.xml;



2) Nesta aula vamos utilizar novamente o arquivo comp para compilação e execução de código java. Caso você opte por baixar/copiar novamente o arquivo, lembre-se que o arquivo inicial que passei para você está com caminhos (paths) de arquivos diferentes do seu ambiente de trabalho e devem ser adaptados novamente;
3) Abra um terminal para digitação de comandos e posicione-se no diretório do tutorial JDBC;
    a. cd mydir/JDBCTutorial
4) Abra o arquivo JDBCUtilities.java para edição com o comando:
    a. gedit src/com/oracle/tutorial/jdbc/JDBCUtilities.java &

```
mury@ubuntu:~$ cd JDBCTutorial/
mury@ubuntu:~/JDBCTutorial$ gedit src/com/oracle/tutorial/jdbc/JDBCUtilities.jav
a &
[1] 16521
mury@ubuntu:~/JDBCTutorial$
```

    b. Certifique-se que este código possui o acréscimo que o permite abrir um banco de dados do postgres, assim como feito na aula anterior:

```
if (this.dbms.equals("mysql")) {
  currentUrlString = "jdbc:" + this.dbms + "://" + this.serverName + ":" + this.portNumber + "/";
  conn = DriverManager.getConnection(currentUrlString, connectionProps);
  this.urlString = currentUrlString + this.dbName;
  conn.setCatalog(this.dbName);
} else if (this.dbms.equals("derby")) {
  this.urlString = "jdbc:" + this.dbms + ":" + this.dbName;
  conn = DriverManager.getConnection(this.urlString + ";create=true", connectionProps);
}
else if (this.dbms.equals("postgresql")) {
  currentUrlString = "jdbc:" + this.dbms + "://" + this.serverName + ":" + this.portNumber + "/" + this.dbName;
  conn = DriverManager.getConnection(currentUrlString, connectionProps);
  this.urlString = currentUrlString + this.dbName;
  conn.setCatalog(this.dbName);
}
System.out.println("Connected to database");
System.out.println(conn);
return conn;
```

5) Pesquise (Ctrl+F) por todas as ocorrências do método printSQLException (definição e uso). Tente entender em que situações esse método é acionado

```
    } catch (SQLException sqle) {
      printSQLException(sqle);
    }
  }

  public static void main(String[] args) {
    JDBCUtilities myJDBCUtilities;
    Connection myConnection = null;
    if (args[0] == null) {
      System.err.println("Properties file not specified at command line");
      return;
    } else {
      try {
        System.out.println("Reading properties file " + args[0]);
        myJDBCUtilities = new JDBCUtilities(args[0]);
      } catch (Exception e) {
        System.err.println("Problem reading properties file " + args[0]);
```

6) Visualize a definição do método printSQLException e identifique qual outro método da classe JDBCUtilities está sendo acionado por este método.

```
public static void printSQLException(SQLException ex) {
  for (Throwable e : ex) {
    if (e instanceof SQLException) {
      if (ignoreSQLException(((SQLException)e).getSQLState()) == false) {
        e.printStackTrace(System.err);
        System.err.println("SQLState: " + ((SQLException)e).getSQLState());
        System.err.println("Error Code: " + ((SQLException)e).getErrorCode());
        System.err.println("Message: " + e.getMessage());
        Throwable t = ex.getCause();
        while (t != null) {
          System.out.println("Cause: " + t);
          t = t.getCause();
        }
      }
    }
  }
}
```

O método chamado aqui proporciona um modo elegante para o tratamento customizado de erros de conexão, em vez de apenas mostrar os erros abre a possibilidade de execução de alguma ação de recuperação sobre os erros.

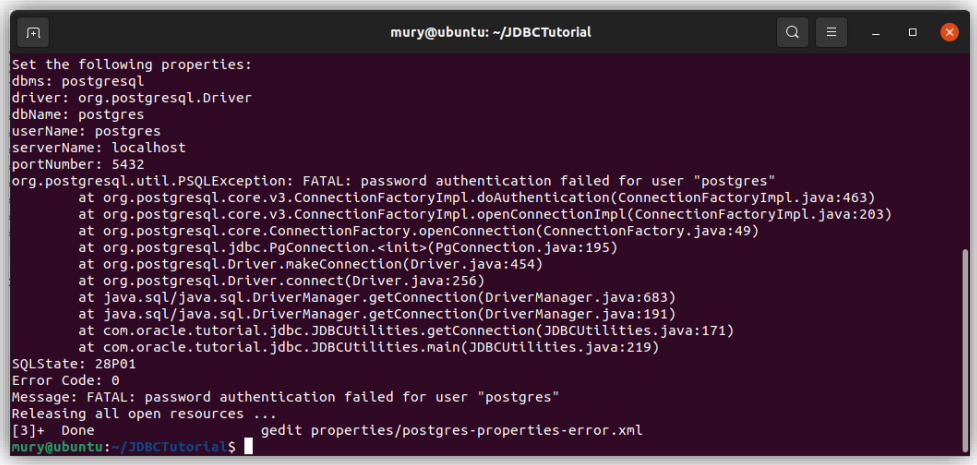
7) Execute o código do JDBCUtilities por meio do script comp, utilizando como parâmetro o arquivo postgres-properties-error.xml.
    a. ./comp JDBCUtilities properties/postgres-properties-error.xml
    b. Como este arquivo ainda é uma cópia do arquivo sem erros nos parâmetros de conexão, então não é esperado outro resultado senão a conexão com sucesso.

```
mury@ubuntu:~/JDBCTutorial$ ./comp JDBCUtilities properties/postgres-properties-error.xml
Reading properties file /home/mury/JDBCTutorial/properties/postgres-properties-error.xml
Set the following properties:
dbms: postgresql
driver: org.postgresql.Driver
dbName: postgres
userName: postgres
serverName: localhost
portNumber: 5432
Connected to database
org.postgresql.jdbc.PgConnection@1ff8b8f
Releasing all open resources ...
mury@ubuntu:~/JDBCTutorial$
```

8) Edite o arquivo postgres-properties-error.xml para inserir erros propositalmente:
    a. gedit properties/postgres-properties-error.xml &
    b. Modifique a senha, por exemplo, deletando uma letra

```
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/mury/JDBCTutorial/BD2-14-postgresql-42.2.4.jar</entry>
7   <entry key="driver">org.postgresql.Driver</entry>
8   <entry key="database_name">postgres</entry>
9   <entry key="user_name">postgres</entry>
10  <entry key="password">123456</entry>
11  <entry key="server_name">localhost</entry>
12  <entry key="port_number">5432</entry>
13
14
15 </properties>
```

    c. Salve o arquivo postgres-properties-error.xml
    d. Execute o passo 7)1. Novamente e observe o resultado; perceba que outros resultados são apresentados na tela.



9) Comente esta linha de código no método printSQLException da classe JDBCUtilities.java e execute o passo 7)1. novamente; compare os resultados:

```
public static void printSQLException(SQLException ex) {
  for (Throwable e : ex) {
    if (e instanceof SQLException) {
      if (ignoreSQLException(((SQLException)e).getSQLState()) == false) {
        //e.printStackTrace(System.err);
        System.err.println("SQLState: " + ((SQLException)e).getSQLState());
        System.err.println("Error Code: " + ((SQLException)e).getErrorCode());
        System.err.println("Message: " + e.getMessage());
        Throwable t = ex.getCause();
        while (t != null) {
          System.out.println("Cause: " + t);
          t = t.getCause();
        }
      }
    }
  }
}
```

```
mury@ubuntu:~/JDBCTutorial$ ./comp JDBCUtilities properties/postgres-properties-error.xml
Reading properties file /home/mury/JDBCTutorial/properties/postgres-properties-error.xml
Set the following properties:
dbms: postgresql
driver: org.postgresql.Driver
dbName: postgres
userName: postgres
serverName: localhost
portNumber: 5432
SQLState: 28P01
Error Code: 0
Message: FATAL: password authentication failed for user "postgres"
Releasing all open resources ...
mury@ubuntu:~/JDBCTutorial$
```

10) Identifique e anote o código de erro alfanumérico que aparece à frente do texto SQLState;

```
mury@ubuntu:~/JDBCTutorial$ ./comp JDBCUtilities properties/postgres-properties-error.xml
Reading properties file /home/mury/JDBCTutorial/properties/postgres-properties-error.xml
Set the following properties:
dbms: postgresql
driver: org.postgresql.Driver
dbName: postgres
userName: postgres
serverName: localhost
portNumber: 5432
SQLState: 28P01
Error Code: 0
Message: FATAL: password authentication failed for user "postgres"
Releasing all open resources ...
mury@ubuntu:~/JDBCTutorial$
```

11) Visualize a definição do método ignoreSQLException pelo editor de texto.

```
public static boolean ignoreSQLException(String sqlState) {
  if (sqlState == null) {
    System.out.println("The SQL state is not defined!");
    return false;
  }
  // X0Y32: Jar file already exists in schema
  if (sqlState.equalsIgnoreCase("X0Y32"))
    return true;
  // 42Y55: Table already exists in schema
  if (sqlState.equalsIgnoreCase("42Y55"))
    return true;
  return false;
}
```

Perceba que existe uma condição inicial para identificar códigos de erros inesperados e duas condições para tratamento de erros específicos. Vamos criar uma condição para tratar o código de erro que identificamos na etapa anterior. A ação corretiva deve ser para instruir o usuário a utilizar um usuário e uma senha padrão de conexão. Nesse caso usaremos o usuário postgres e a senha postgres como exemplos, mas poderia ser um outro usuário com menos privilégios de acesso ao banco de dados;

12) Modifique este método para tratar o erro de login:
    a. Copie e cole o código da condição 42Y55 logo abaixo, mas antes do return false;
    b. O código a ser executado nesta condição é imprimir uma mensagem com um usuário padrão e sua senha de conexão ao banco de dados.
    c. Salve JDBCUtilities.java, compile e execute-o novamente;

```
public static boolean ignoreSQLException(String sqlState) {
  if (sqlState == null) {
    System.out.println("The SQL state is not defined!");
    return false;
  }
  // X0Y32: Jar file already exists in schema
  if (sqlState.equalsIgnoreCase("X0Y32"))
    return true;
  // 42Y55: Table already exists in schema
  if (sqlState.equalsIgnoreCase("42Y55"))
    return true;
  if (sqlState.equalsIgnoreCase("28P01"))
      System.out.println("Senha e Usuario incorretos! Use postgres nos respectivos campos.");
  return false;
}
```

```
mury@ubuntu:~/JDBCTutorial$ ./comp JDBCUtilities properties/postgres-properties-error.xml
Reading properties file /home/mury/JDBCTutorial/properties/postgres-properties-error.xml
Set the following properties:
dbms: postgresql
driver: org.postgresql.Driver
dbName: postgres
userName: postgres
serverName: localhost
portNumber: 5432
Senha e Usuario incorretos! Use postgres nos respectivos campos.
SQLState: 28P01
Error Code: 0
Message: FATAL: password authentication failed for user "postgres"
Releasing all open resources ...
mury@ubuntu:~/JDBCTutorial$
```

13) Simule erros de conexão para os demais parâmetros de conexão do arquivo postgresproperties-error.xml, identifique os códigos alfanuméricos de erros retornados pelo SQLState e crie condições customizadas para instruir um usuário do seu programa sobre como proceder;

a) Segue os tratamentos e códigos de erros para cada um dos campos do arquivo postgres-properties-error.xml

```
public static boolean ignoreSQLException(String sqlState) {
  if (sqlState == null) {
    System.out.println("The SQL state is not defined!");
    return false;
  }
  // X0Y32: Jar file already exists in schema
  if (sqlState.equalsIgnoreCase("X0Y32"))
    return true;
  // 42Y55: Table already exists in schema
  if (sqlState.equalsIgnoreCase("42Y55"))
    return true;
  if (sqlState.equalsIgnoreCase("28P01"))
      System.out.println("Senha e Usuario incorretos! Use postgres nos respectivos campos.");
  if (sqlState.equalsIgnoreCase("3D000"))
      System.out.println("Nome do Banco de Dados incorreto!");
  if (sqlState.equalsIgnoreCase("08001"))
      System.out.println("Nome do servidor ou porta incorreto!");
  return false;
}
```

b) Segue o erro quando o servidor ou porta está incorreto.

```
mury@ubuntu:~/JDBCTutorial$ ./comp JDBCUtilities properties/postgres-properties-error.xml
Reading properties file /home/mury/JDBCTutorial/properties/postgres-properties-error.xml
Set the following properties:
dbms: postgresql
driver: org.postgresql.Driver
dbName: postgres123
userName: postgres123
serverName: localhost123
portNumber: 5432123
Jul 21, 2022 3:25:56 PM org.postgresql.Driver parseURL
WARNING: JDBC URL port: 5432123 not valid (1:65535)
Nome do servidor ou porta incorreto!
SQLState: 08001
Error Code: 0
Message: No suitable driver found for jdbc:postgresql://localhost123:5432123/postgres123
Releasing all open resources ...
mury@ubuntu:~/JDBCTutorial$
```

c) Segue o erro quando o nome do Banco de Dados está incorreto.

```
mury@ubuntu:~/JDBCTutorial$ ./comp JDBCUtilities properties/postgres-properties-error.xml
Reading properties file /home/mury/JDBCTutorial/properties/postgres-properties-error.xml
Set the following properties:
dbms: postgresql
driver: org.postgresql.Driver
dbName: postgres123
userName: postgres
serverName: localhost
portNumber: 5432
Nome do Banco de Dados incorreto!
SQLState: 3D000
Error Code: 0
Message: FATAL: database "postgres123" does not exist
Releasing all open resources ...
mury@ubuntu:~/JDBCTutorial$
```

d) O erro para Usuário e Senha Incorretos foi demonstrado no passo 12.

14) Introduza ao menos dois tipos de erros no código SQL das classes MyQueries*1 (aula anterior) identifique os códigos alfanuméricos de erros retornados pelo SQLState e crie condições customizadas no JDBCUtilities.java para instruir um usuário do seu programa sobre como proceder;

a) Nesse caso, digitei o nome da tabela errado.

```
public static void getMyData(Connection con) throws SQLException {
  Statement stmt = null;
  String query =
    "select d.nome_cliente, sum(d.saldo_deposito)" +
    "from maer d where d.nome_cliente not in" +
      "(select e.nome_cliente from emprestimo e)" +
    "group by d.nome_cliente;";
  try {
```

```
mury@ubuntu:~/JDBCTutorial$ ./comp MyQueries2 properties/postgres-properties-error.xml
Set the following properties:
dbms: postgresql
driver: org.postgresql.Driver
dbName: postgres
userName: postgres
serverName: localhost
portNumber: 5432
Connected to database
org.postgresql.jdbc.PgConnection@387c703b
SQLState: 42P01
Error Code: 0
Message: ERROR: relation "maer" does not exist
  Position: 50
Releasing all open resources ...
mury@ubuntu:~/JDBCTutorial$
```

b) Nesse caso digitei o nome da coluna errado.

```java
public static void getMyData(Connection con) throws SQLException {
  Statement stmt = null;
  String query =
    "select d.client_name, sum(d.saldo_deposito)" +
    "from deposito d where d.nome_cliente not in" +
      "(select e.nome_cliente from emprestimo e)" +
    "group by d.nome_cliente;";
  try {
    stmt = con.createStatement();
    ResultSet rs = stmt.executeQuery(query);
    System.out.println("Soma dos depositos dos clientes");
```

```
mury@ubuntu:~/JDBCTutorial$ ./comp MyQueries2 properties/postgres-properties-error.xml
Set the following properties:
dbms: postgresql
driver: org.postgresql.Driver
dbName: postgres
userName: postgres
serverName: localhost
portNumber: 5432
Connected to database
org.postgresql.jdbc.PgConnection@387c703b
SQLState: 42703
Error Code: 0
Message: ERROR: column d.client_name does not exist
  Position: 8
Releasing all open resources ...
```

c) Segue os tratamentos e códigos de erros para esses casos

```java
public static boolean ignoreSQLException(String sqlState) {
  if (sqlState == null) {
    System.out.println("The SQL state is not defined!");
    return false;
  }
  // X0Y32: Jar file already exists in schema
  if (sqlState.equalsIgnoreCase("X0Y32"))
    return true;
  // 42Y55: Table already exists in schema
  if (sqlState.equalsIgnoreCase("42Y55"))
    return true;
  if (sqlState.equalsIgnoreCase("28P01"))
      System.out.println("Senha e Usuario incorretos! Use postgres nos respectivos campos.");
  if (sqlState.equalsIgnoreCase("3D000"))
      System.out.println("Nome do Banco de Dados incorreto!");
  if (sqlState.equalsIgnoreCase("08001"))
      System.out.println("Nome do servidor ou porta incorreto!");
  if (sqlState.equalsIgnoreCase("42703"))
      System.out.println("Coluna seleciona nao existe!");
  if (sqlState.equalsIgnoreCase("42P01"))
      System.out.println("Tabela seleciona nao existe!");
  return false;
}
```