

Conectado o Java ao banco de dados

Primeiramente, baixe e instale o SQLite Portable a partir do link a seguir:

https://download.sqlitebrowser.org/SQLiteDatabaseBrowserPortable_3.11.2_English.paf.exe

Em seguida, baixe os drivers necessários nos links abaixo:

Drivers:

<https://github.com/xerial/sqlite-jdbc/releases/download/3.46.0.0/sqlite-jdbc-3.46.0.0.jar>

<https://search.maven.org/remotecontent?filepath=org/slf4j/slf4j-api/1.7.36/slf4j-api-1.7.36.jar>

<https://repo1.maven.org/maven2/org/slf4j/slf4j-simple/1.7.36/slf4j-simple-1.7.36.jar>

→ Adicione os drivers em uma mesma pasta para facilitar seu trabalho.

Agora, vamos adicionar os drivers ao nosso projeto:

- Clique com o botão direito no seu projeto.
- Vá para "Open Module Settings".
- Selecione "Libraries".
- Clique no ícone de "+" e escolha "Java".
- Navegue até o local onde você baixou o arquivo JAR do driver e selecione-o.
- Vá para "Modules".
- Selecione o seu módulo, vá para a aba "Dependencies" e adicione o JAR dos drivers.

→ Para configurar nosso projeto:

Crie a classe Clientes com o seguinte código:

```
import java.sql.*;

public class Clientes {
    private static final String URL = "jdbc:sqlite:BancoDeDados.db";
    private Connection connection;

    public void connect() {
        try {
            Class.forName("org.sqlite.JDBC");
            connection = DriverManager.getConnection(URL);
            connection.setAutoCommit(false);
            System.out.println("Conexão realizada! Banco de dados: " + URL);
        } catch (ClassNotFoundException e) {
            System.out.println("Driver JDBC do SQLite não encontrado: " + e.getMessage());
        } catch (SQLException e) {
            System.out.println("Erro ao conectar: " + e.getMessage());
        }
    }

    public void close() {
        if (connection != null) {
            try {
```

```

        connection.close();
        System.out.println("Conexão fechada.");
    } catch (SQLException e) {
        System.out.println("Erro ao fechar a conexão: " + e.getMessage());
    }
}

}

}

public void createTable() {
    try (Statement statement = connection.createStatement()) {
        statement.execute("CREATE TABLE IF NOT EXISTS Clientes (ID INTEGER PRIMARY KEY, Nome VARCHAR)");
        connection.commit();
        System.out.println("Tabela criada ou já existe.");
    } catch (SQLException e) {
        System.out.println("Erro ao criar tabela: " + e.getMessage());
    }
}

public void insertClient(String nome) {
    try (PreparedStatement insertStatement = connection.prepareStatement("INSERT INTO Clientes (Nome) VALUES (?)")) {
        insertStatement.setString(1, nome);
        insertStatement.executeUpdate();
        connection.commit();
        System.out.println("Cliente inserido.");
    } catch (SQLException e) {
        System.out.println("Erro ao inserir cliente: " + e.getMessage());
    }
}

public void queryClients() {
    try (PreparedStatement selectStatement = connection.prepareStatement("SELECT * FROM Clientes")) {
        ResultSet resultSet = selectStatement.executeQuery();
        System.out.println("Clientes:");
        while (resultSet.next()) {
            int id = resultSet.getInt("ID");
            String nome = resultSet.getString("Nome");
            System.out.println("Cliente ID: " + id + ", Nome: " + nome);
        }
    } catch (SQLException e) {
        System.out.println("Erro ao consultar clientes: " + e.getMessage());
    }
}

public void updateClient(int id, String novoNome) {
    try (PreparedStatement updateStatement = connection.prepareStatement("UPDATE Clientes SET Nome = ? WHERE ID = ?")) {
        updateStatement.setString(1, novoNome);
        updateStatement.setInt(2, id);
        updateStatement.executeUpdate();
        connection.commit();
        System.out.println("Cliente atualizado.");
    } catch (SQLException e) {
        System.out.println("Erro ao atualizar cliente: " + e.getMessage());
    }
}

public void deleteClient(int id) {
    try (PreparedStatement deleteStatement = connection.prepareStatement("DELETE FROM Clientes WHERE ID = ?")) {
        deleteStatement.setInt(1, id);
        deleteStatement.executeUpdate();
        connection.commit();
        System.out.println("Cliente deletado.");
    }
}

```

```
    } catch (SQLException e) {  
        System.out.println("Erro ao deletar cliente: " + e.getMessage());  
    }  
}  
}
```

Na classe Main(), crie um objeto do tipo Clientes e faça a chamada dos métodos, sempre iniciando com a conexão com o banco de dados e a criação da tabela antes de chamar os outros métodos.

Agora é com você:

Altere esse programa do banco de dados, adicionando classes, se necessário, para que ele possa executar as funcionalidades do programa feito na última aula.