Java Database Connectivity (JDBC)

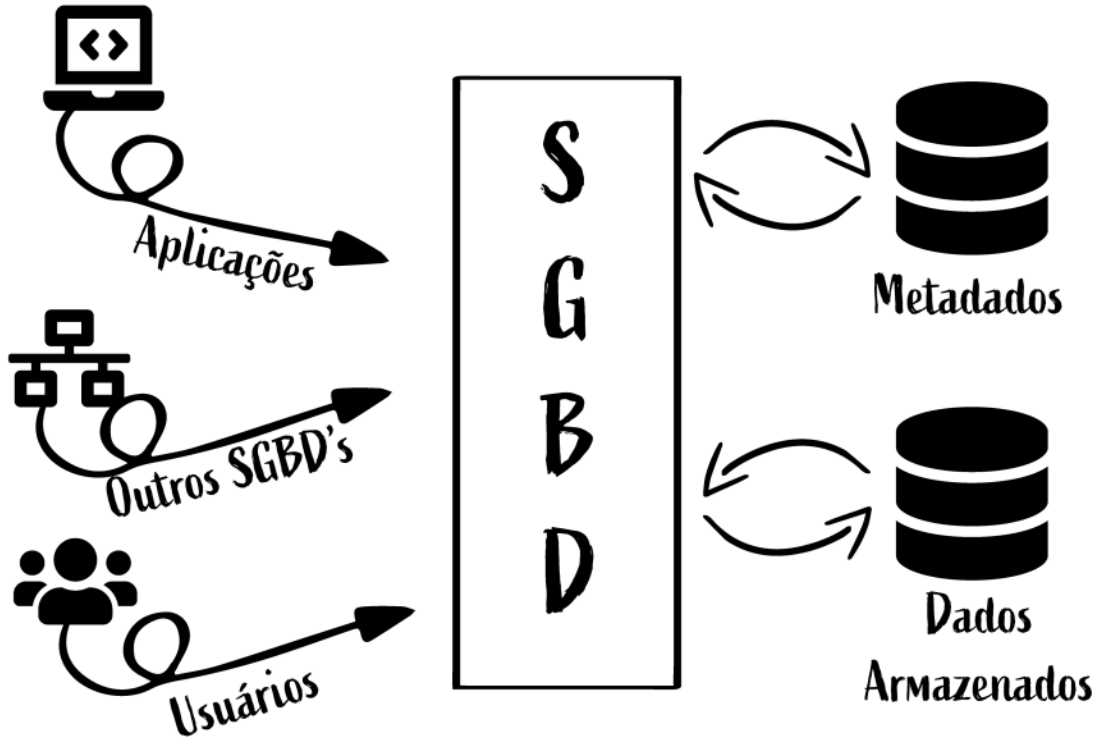
Prof. Danilo Rodrigues Pereira

danilo.pereira84@gmail.com

Sistema de Gerenciamento de Banco de Dados - SGBD

- Os SGBD são os elementos mais comuns para persistência de dados utilizados em aplicações comerciais, pois propiciam formas padronizadas para inserção, alteração, remoção e busca de dados.

- Portanto, é necessário verificar como as interfaces gráficas, quando acionadas pelo usuário, fazem o uso dos SGDBs para gravar seus dados



Sistema de Gerenciamento de Banco de Dados - SGBD

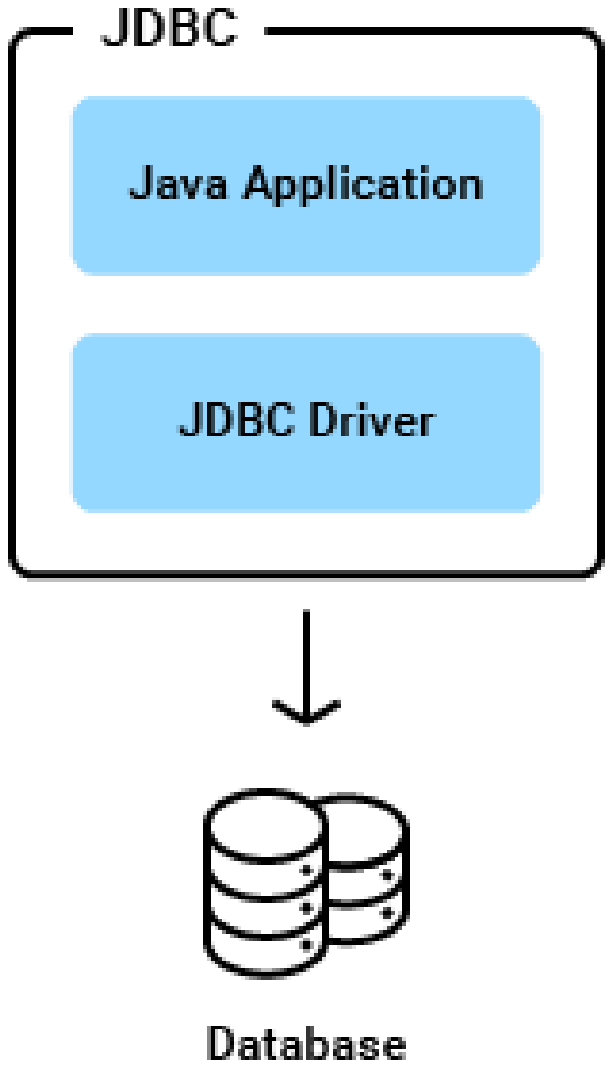
- Para utilizar os SGDBs em Java, especialmente em interfaces gráficas em Java Swing, é podemos utilizar o **Java Database Connectivity (JDBC)**.

- O JDBC consiste em um conjunto de classes que são incorporadas ao Java Development Kit (JDK)

Java Database Connectivity (JDBC)

- Para possibilitar o acesso a diversos SGDBs de forma padronizada sem a necessidade de se utilizar formas específicas para cada sistema de banco de dados.

- O JDBC é compatível com diversos sistemas de banco de dados, tais como:
  - MySQL
  - Oracle
  - PostgreSQL
  - SQLite

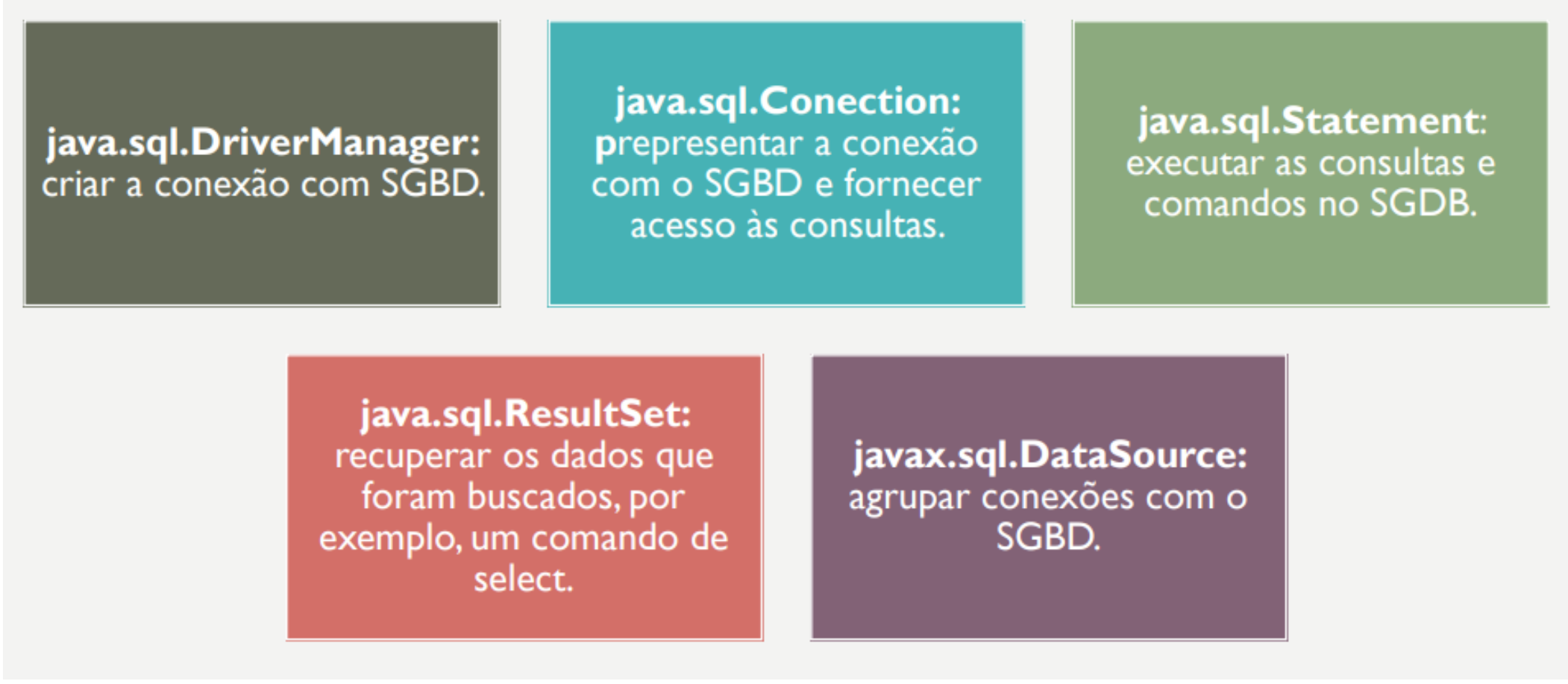


Principais falhas

- Banco de Dados não iniciado;
- Driver não Encontrado
- Driver incompatível;
- URL – Configurada de forma incorreta (nome banco, usuário, senha, servidor)
- Classe de conexão (inexistente, não instanciada, com erros)
- Usuário sem direito de acesso –
- Falta de tratamento de Exceções try .... catch

Arquitetura JDBC

- O JDBC utiliza as classes que controlam o driver que será utilizado para se conectar no banco de dados indicado. O primeiro passo, **que consiste em estabelecer a conexão, é necessário garantir que o JDBC conheça o SGBD**



String de conexão do Banco de Dados (URL)

- Para se conectar a um banco de dados, esteja ele implementado em qualquer SGBD, é necessário criar uma string de conexão, ou URL JDBC (Uniform Resource Locator JDBC).

- Essa string informará o “caminho” do banco e apresenta a seguinte sintaxe:

jdbc:<driver>:<detalhes da conexão>

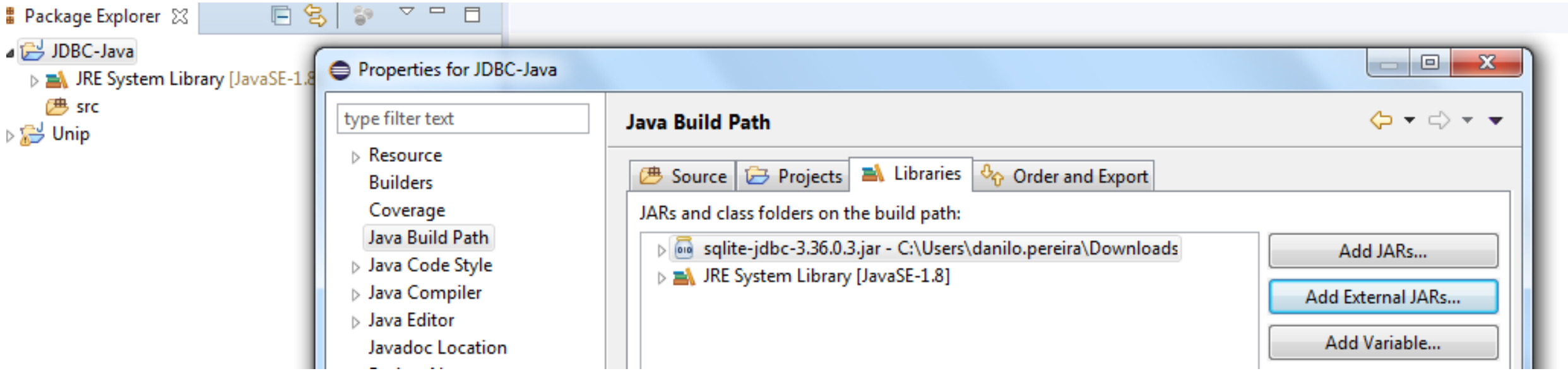
String de conexão do Banco de Dados (URL) - Exemplos

Banco de dados	URL JDBC
MySQL	jdbc:mysql://localhost:3306/nomeBancoDeDados
SQL Server	jdbc:sqlserver://localhost;databaseName=nomeBancoDeDados
Oracle	jdbc:oracle:thin@myserver:1521:nomeBancoDeDados

Usando SQLite em Java

1) Baixar o jar do SQLite - https://mvnrepository.com/artifact/org.xerial/sqlite-jdbc/3.36.0.3

2) Adicionar o jar no projeto



Verificando os drivers disponível para o seu projeto Java

Antes de iniciar a criação da classe para conexão com o banco de dados, é extremamente importante garantir que o seu projeto faz o carregamento correto dos drives JDBC

Para isso, vamos criar uma classe Java, que exibirá uma lista de todos os drivers dos diferentes fabricantes (Oracle, MySQL, IBM, Postgresql, etc)

Verificando os drives disponíveis para o seu projeto Java

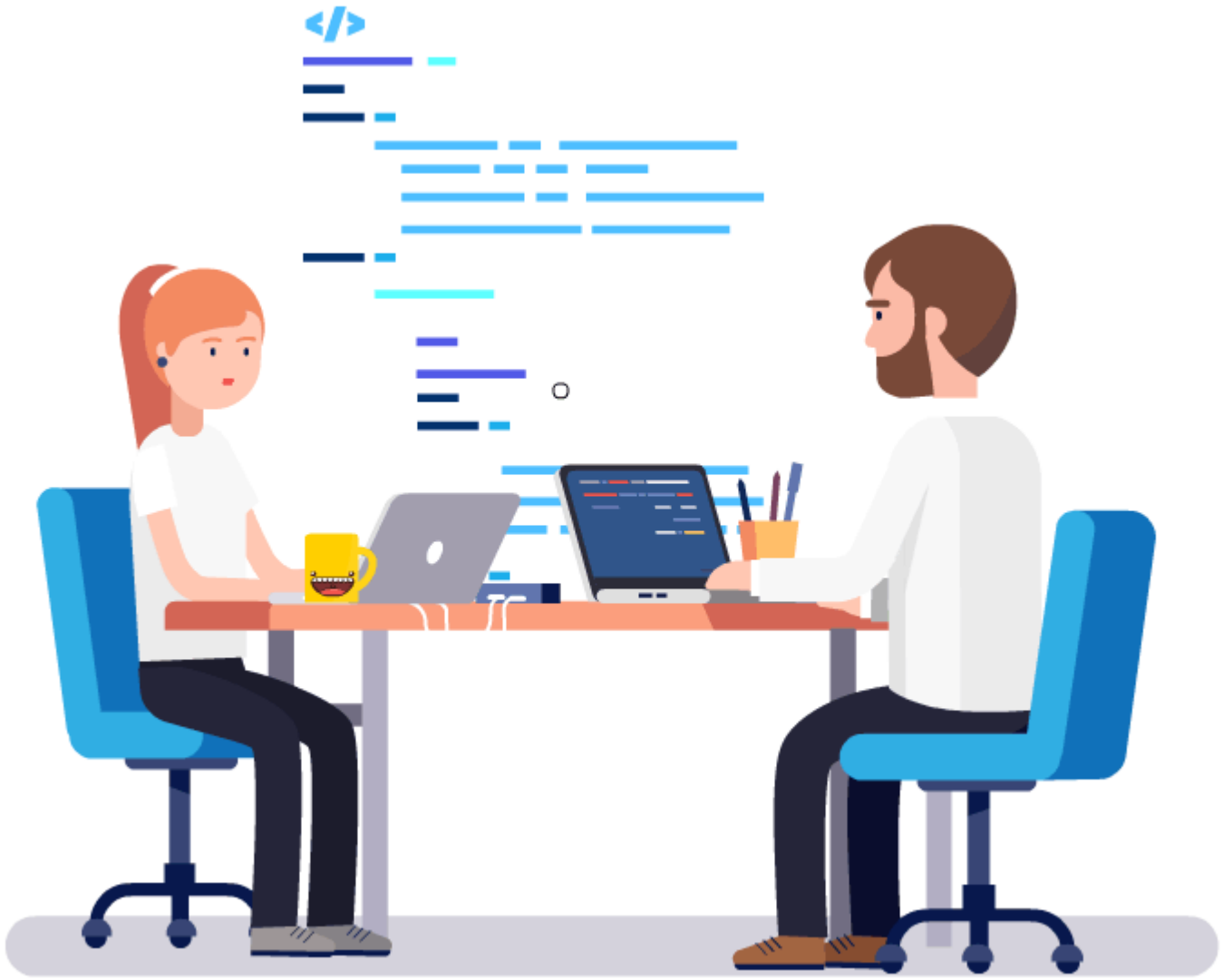
```
import java.sql.Driver;
import java.sql.DriverManager;
import java.util.Enumeration;

public class PrintJDBCDrivers {

     public static void main(String[] args) {
          System.out.println("List of loaded JDBC drivers");
          for (Enumeration<Driver> e = DriverManager.getDrivers(); e.hasMoreElements();) {
               Driver driver = e.nextElement();
               print(driver);
          }
     }

     public static void print(Driver driver) {
          String className = driver.getClass().getName();
          int majorVersion = driver.getMajorVersion();
          int minorVersion = driver.getMinorVersion();
          System.out.println("---------------------------------------");
          System.out.println("Name Driver: " + className);
          System.out.println("Driver Major Version: " + majorVersion);
          System.out.println("Driver Minor Version: " + minorVersion);
          System.out.println("---------------------------------------");
     }
}
```

VAMOS PRATICAR ?



Conectando no banco de dados - Exemplo

```
public class Connect {

     public static void connect() {
          Connection conn = null;
          try {
               String url = "jdbc:sqlite:database_test.db";

               conn = DriverManager.getConnection(url);

               System.out.println("Connection to SQLite has been established.");

          } catch (SQLException e) {
               System.out.println(e.getMessage());
          } finally {
               try {
                    if (conn != null) {
                    conn.close();
                    }
               } catch (SQLException ex) {
               System.out.println(ex.getMessage());
               }
          }
     }

     public static void main(String[] args) {
         connect();
     }
}
```

Criando TABELAS

```
public class CreateTable {

    public static void createNewTable() {
        String url = "jdbc:sqlite:database_test.db";

        StringBuffer sql = new StringBuffer();
        sql.append("CREATE TABLE IF NOT EXISTS cliente (");
        sql.append("id integer PRIMARY KEY , ");
        sql.append("nome text NOT NULL, ");
        sql.append("idade integer, ");
        sql.append("cpf text NOT NULL, ");
        sql.append("rg text ");
        sql.append(")");

        try (Connection conn = DriverManager.getConnection(url);
          Statement stmt = conn.createStatement()) {
          stmt.execute(sql.toString());
        } catch (SQLException e) {
        System.out.println(e.getMessage());
        }
    }

    public static void main(String[] args) {
          createNewTable();
    }
}
```

Inserindo dados - Exemplo

```
public class InsertClient {

private Connection connect() {
        // SQLite connection string
        String url = "jdbc:sqlite:database_test.db";
        Connection conn = null;
        try {
            conn = DriverManager.getConnection(url);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
        return conn;
    }

    public void insert(Cliente cliente) {
        String sql = "INSERT INTO cliente(nome,idade, cpf, rg) VALUES(?,?,?,?)";

        try (Connection conn = this.connect();
                PreparedStatement pstmt = conn.prepareStatement(sql)) {
            pstmt.setString(1, cliente.getNome());
            pstmt.setInt(2, cliente.getIdade());
            pstmt.setString(3, cliente.getCpf());
            pstmt.setString(4, cliente.getRg());
            pstmt.executeUpdate();
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

```
public static void main(String[] args) {
        InsertClient newClient = new InsertClient();

        Cliente c1 = new Cliente();
        c1.setNome("Danilo R. Pereira");
        c1.setIdade(37);
        c1.setCpf("111.111.111-11");
        c1.setRg("222.222.222-22");

        newClient.insert(c1);

}
```

Selecionando dados - Exemplo

```
public class SelectClient {

	private Connection connect() {
		String url = "jdbc:sqlite:database_test.db";
		Connection conn = null;
		try {
		conn = DriverManager.getConnection(url);
		} catch (SQLException e) {
		System.out.println(e.getMessage());
		}
		return conn;
	}

	public void selectAll() {
		String sql = "SELECT id, nome, idade, cpf, rg FROM cliente";
		try (Connection conn = this.connect();
		Statement stmt = conn.createStatement();
		ResultSet rs = stmt.executeQuery(sql)) {
			while (rs.next()) {
				System.out.println(rs.getInt("id") + "\t" + rs.getString("nome") + "\t" + rs.getInt("idade") + "\t" +
				rs.getString("cpf") + "\t" + rs.getString("rg"));
				}
			} catch (SQLException e) {
			System.out.println(e.getMessage());
			}
	}

	public static void main(String[] args) {
		SelectClient app = new SelectClient();
		app.selectAll();
	}
}
```

Atualizando dados - Exemplo

```
public class UpdateCliente {

    private Connection connect() {
        String url = "jdbc:sqlite:database_test.db";
        Connection conn = null;
        try {
            conn = DriverManager.getConnection(url);
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
        return conn;
    }

    public void update(int id, String name, Integer idade) {
        String sql = "UPDATE cliente SET nome = ? , + "idade = ? " + "WHERE id = ?";
        try (Connection conn = this.connect();
            PreparedStatement pstmt = conn.prepareStatement(sql)) {
            pstmt.setString(1, name);
            pstmt.setInt(2, idade);
            pstmt.setInt(3, id);
           pstmt.executeUpdate();
        } catch (SQLException e) {
            System.out.println(e.getMessage());
        }
    }

    public static void main(String[] args) {
        UpdateCliente app = new UpdateCliente();
        app.update(1, "Danilo Rodrigues Pereira", 38);
    }
}
```