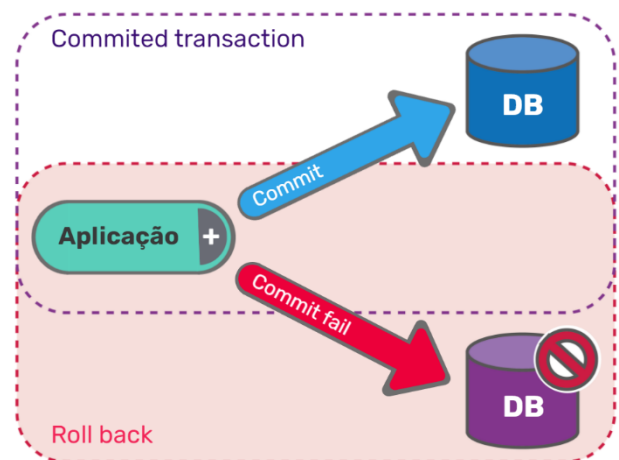




Certified Tech Developer
The Ultimate Degree

Transações e Consultas

```
try {
    ....
    connection.setAutoCommit(false);
    statement.executeUpdate("INSERT INTO RESERVAS(Passageiro,
    Origem, Destino) VALUES('Felipe', 'São Paulo', 'Madrid')");
    statement.executeUpdate("INSERT INTO RESERVAS(Passageiro,
    Origem, Destino) VALUES('Felipe', 'Madrid', 'Dubai')");
    statement.executeUpdate("INSERT INTO RESERVAS(Passageiro,
    Origem, Destino) VALUES('Felipe', 'Dubai', 'Sydney')");
    connection.commit();
    ....
} catch(SQLException e) {
    if(connection!=null){
        try {
            connection.rollback();
        } catch (SQLException ex) {
            System.out.println(ex.toString());
        }
    }
}
```



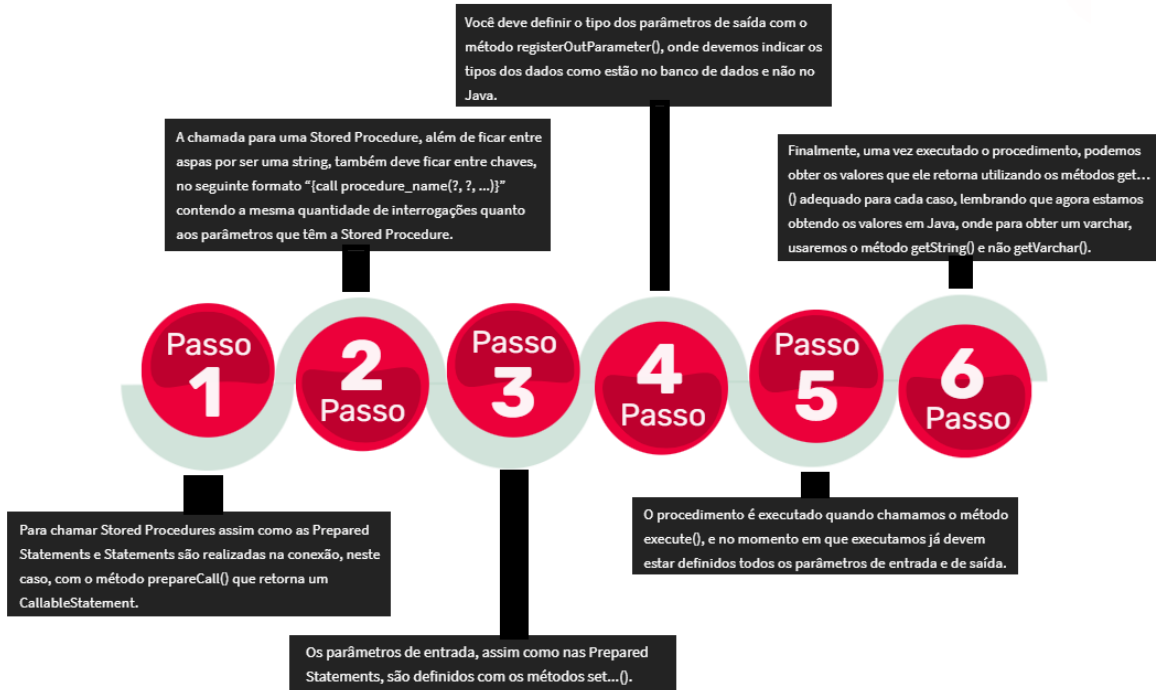
Passos para invocar Stored Procedures

Agora chegou o momento de dar algumas dicas e passos para chamar um procedimento armazenado (Stored Procedure) em JDBC.



Passos para invocar Stored Procedures

Vamos descobrir o que acontece a cada momento, passando o mouse sobre o número



```
public static void main(String[] args) {
    BufferedReader input = new BufferedReader(new InputStreamReader(System.in));
    int id = -1;
    Connection cn = null;

    try {
        // Carrega o driver
        Class.forName("com.mysql.jdbc.Driver");
        DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb", "root", "myPassword");

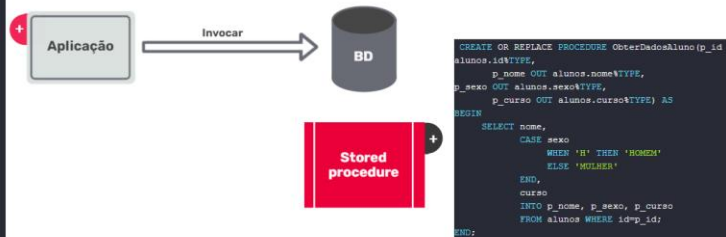
        // Instrução para chamada da Stored Procedure
        CallableStatement cst = cn.prepareCall("{call obterDadosAluno(?, ?, ?, ?)}");

        do {
            System.out.println("Informe o ID do aluno:");
            try {
                id = Integer.parseInt(input.readLine());
            } catch (IOException ex) {
                System.out.println("Erro...");
            }

            // Parâmetro 1 da Stored Procedure
            cst.setInt(1, id);

            // Definimos os tipos de parâmetros de saída da Stored Procedure
            cst.registerOutParameter(2, java.sql.Types.VARCHAR);
            cst.registerOutParameter(3, java.sql.Types.VARCHAR);
            cst.registerOutParameter(4, java.sql.Types.VARCHAR);

            // Executa a Stored Procedure
            cst.execute();
        } while (id != -1);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```



```
public static void main(String[] args) {
    BufferedReader input = new BufferedReader(new InputStreamReader(System.in));
    int id = -1;
    Connection cn = null;

    try {
        // Carregue o driver
        Connection cn =
        DriverManager.getConnection("jdbc:h2:" + "./Database/my", "root", "myPassword");

        // Instrução para chamada da Stored Procedure
        CallableStatement cst = cn.prepareCall("{call ObterDadosAluno (?, ?, ?, ?)}");

        do {
            System.out.println("\nInforme o ID do aluno:");
            try {
                id = Integer.parseInt(entrada.readLine());
            } catch (IOException ex) {
                System.out.println("Erro...");
            }

            // Parâmetro 1 da Stored Procedure
            cst.setInt(1, id);

            // Definimos os tipos de parâmetro de saída da Stored Procedure
            cst.registerOutParameter(2, java.sql.Types.VARCHAR);
            cst.registerOutParameter(3, java.sql.Types.VARCHAR);
            cst.registerOutParameter(4, java.sql.Types.VARCHAR);

            // Executa a Stored Procedure
            cst.execute();

            // Se obtém o retorno da Stored Procedure
            String nome = cst.getString(2);
            String sexo = cst.getString(3);
            String curso = cst.getString(4);
            System.out.println("Nome: " + nome);
            System.out.println("Sexo: " + sexo);
            System.out.println("Curso: " + curso);
        } while (id > 0);

    } catch (SQLException ex) {
        System.out.println("Erro: " + ex.getMessage());
    } finally {
        try {
            cn.close();
        } catch (SQLException ex) {
            System.out.println("Erro: " + ex.getMessage());
        }
    }
}
```

```
CREATE OR REPLACE PROCEDURE ObterDadosAluno(p_id alunos.id%TYPE,
    p_nome OUT alunos.nome%TYPE, p_sexo OUT alunos.sexo%TYPE,
    p_curso OUT alunos.curso%TYPE) AS
BEGIN
    SELECT nome,
        CASE sexo
            WHEN 'H' THEN 'HOMEM'
            ELSE 'MULHER'
        END,
        curso
    INTO p_nome, p_sexo, p_curso
    FROM alunos WHERE id=p_id;

END;
```