

## Diferenças entre Statement e PreparedStatement

**PreparedStatement** --> **Prepared** --> É mais rápido que o Statement, quando você precisa executar várias vezes a mesma coisa, mudando por exemplo apenas os parâmetros, pois ele "prepara" o acesso e com isso se for fazer, por exemplo, 5 inserts, será muito mais rápido que o Statement

Quando algo chega para ser executado no BD, ocorrem diversos passos de preparação até a execução e utilizando o PreparedStatement esses passos de preparação ocorrem apenas uma vez, já no Statement ocorre em toda a execução.

Vamos ver um exemplo: imagine que você quer fazer as seguintes consultas:

```
1 | insert into usuarios (nome, idade) values ('Amanda', 35);
2 | insert into usuarios (nome, idade) values ('Bruno', 22);
3 | insert into usuarios (nome, idade) values ('Carlos', 29);
```

## Statement

Você pode criar um Statement e chamar o método executeUpdate para cada uma das consultas, assim:

```
Statement stmt = conn.createStatement();

stmt.executeUpdate("insert into usuarios (nome, idade) values ('Amanda', 35)");
stmt.executeUpdate("insert into usuarios (nome, idade) values ('Bruno', 22)");
stmt.executeUpdate("insert into usuarios (nome, idade) values ('Carlos', 29)");
```

## PreparedStatement

Ou, então, criar um PreparedStatement que será enviado para o banco de dados e que, por sua vez, irá fazer todos os passos de preparação antes da execução definitiva. Logo após, basta mudar os valores das variáveis (?) e enviá-las para execução no banco, assim:

```
PreparedStatement stmt = conn.prepareStatement("insert into usuarios (nome, idade) values (?, ?)");
```

```
stmt.setString(1, "Amanda");  
stmt.setString(2, "35");  
stmt.executeUpdate();
```

```
stmt.setString(1, "Bruno");  
stmt.setString(2, "22");  
stmt.executeUpdate();
```

```
stmt.setString(1, "Carlos");  
stmt.setString(2, "29");  
stmt.executeUpdate();
```

Lógico que você não irá notar muita diferença de velocidade ao enviar apenas três consultas para o banco de dados, mas se você estiver fazendo isso muitas vezes, o PreparedStatement pode ajudar.

Fazendo um teste com a inserção de 1.000.000 usuários em uma tabela similar à do exemplo, irá obter os seguintes resultados:

TIPO	TEMPO TOTAL (MS)	TEMPO TOTAL (MIN)	TEMPO POR CONSULTA (MS)
STATEMENT	714386	11,9	0,714
PREPAREDSTATEMENT	657988	10,9	0,658

Isso significa que nesse teste o PreparedStatement foi aproximadamente 7,9% mais rápido que o Statement!

## **Conclusão**

Se você for fazer muitas consultas onde apenas os valores mudam, use o PreparedStatement.