

## Sistemas de Bancos de Dados

### Aula 13 – Comando pré-compilados

Murielly Oliveira Nascimento – 11921BSI222

1) Inicialmente crie um método no seu programa java (MyQueries, por exemplo) com o código abaixo, inclua a sua chamada no método main e teste a execução:

```
public static void insertMyData1(Connection con) throws SQLException {
    Statement stmt = null;
    String query = null;
    query = "insert into debito (numero_debito, valor_debito, motivo_debito, data_debito, numero_conta, nome_agencia, nome_cliente)
    values (3000,3000,5,'2014-02-06',36593,'UFU','Pedro Alvares Sousa');" ;
    try {
        stmt = con.createStatement();
        stmt.executeUpdate(query);
        if (stmt != null) { stmt.close(); }
        System.out.println("Debitos da Instituicao Bancaria atualizados.");
    } catch (SQLException e) {
        JDBCUtilities.printSQLException(e);
    }
}
```

```
mury@ubuntu:~/JDBCTutorial$ ./comp MyQueries properties/postgres-properties.xml
Set the following properties:
dbms: postgresql
driver: org.postgresql.Driver
dbName: IB2
userName: postgres
serverName: localhost
portNumber: 5432
Connected to database
org.postgresql.jdbc.PgConnection@387c703b
Debitos da Instituicao Bancaria atualizados.
Releasing all open resources ...
mury@ubuntu:~/JDBCTutorial$
```

2) Agora faça uma cópia do método insertMyData1 para insertMyData2 e modifique está cópia para utilizar comandos pré-compilados. Para tanto faça:

```
public static void insertMyData2(Connection con) throws SQLException {
    PreparedStatement stmt = null;
    String query = null;
    query = "insert into debito (numero_debito, valor_debito, motivo_debito, data_debito, numero_conta, nome_agencia, nome_cliente)
    values (?, ?, ?, ?, ?, ?, ?)";
    try {
        stmt = con.prepareStatement(query);
        stmt.setInt(1, 3001); stmt.setDouble(2, 3001); stmt.setInt(3, 4);
        stmt.setDate(4, Date.valueOf("2014-02-06")); stmt.setInt(5, 36593);
        stmt.setString(6, "UFU"); stmt.setString(7, "Pedro Alvares Sousa");
        stmt.executeUpdate();
        if (stmt != null) { stmt.close(); }
        System.out.println("Debitos da Instituicao Bancaria atualizados.");
    } catch (SQLException e) {
        JDBCUtilities.printSQLException(e);
    }
}
```

```

mury@ubuntu:~/JDBCTutorial$ ./comp MyQueries properties/postgres-properties.xml
Set the following properties:
dbms: postgresql
driver: org.postgresql.Driver
dbName: IB2
userName: postgres
serverName: localhost
portNumber: 5432
Connected to database
org.postgresql.jdbc.PgConnection@387c703b
Debitos da Instituicao Bancaria atualizados.
Releasing all open resources ...
mury@ubuntu:~/JDBCTutorial$

```

IB2/postgres@localhost

Query Editor Query History

```

1  -- Database: IB2
2
3  -- DROP DATABASE IF EXISTS "IB2";
4
5  -- CREATE DATABASE "IB2"
6  --   WITH
7  --   OWNER = postgres
8  --   ENCODING = 'UTF8'
9  --   LC_COLLATE = 'en_US.UTF-8'
10 --   LC_CTYPE = 'en_US.UTF-8'
11 --   TABLESPACE = pg_default
12 --   CONNECTION LIMIT = -1;
13
14 select * from debito where numero_conta = 36593 and nome_agencia = 'UFU' and
15 nome_cliente = 'Pedro Alvares Sousa';
16

```

Data Output Explain Messages Notifications

	numero_debito [PK] integer	valor_debito double precision	motivo_debito smallint	data_debito date	numero_conta integer	nome_agencia character varying	nome_cliente character varying
1		1102	1131.48	2	2013-10-02	36593 UFU	Pedro Alvares Sousa
2		1193	1095.51	5	2012-04-16	36593 UFU	Pedro Alvares Sousa
3		1292	545.42	3	2012-12-16	36593 UFU	Pedro Alvares Sousa

3) Agora vamos medir o tempo de execução entre os comandos Statement e PreparedStatement. Para tanto devemos inserir o seguinte trecho de código nos métodos insertMyData1 e insertMyData2. Lembre-se que, para executarmos o código com as novas alterações, o método main agora deve chamar ambos os métodos.

```

mury@ubuntu:~/JDBCTutorial$ ./comp MyQueries properties/postgres-properties.xml
Set the following properties:
dbms: postgresql
driver: org.postgresql.Driver
dbName: IB2
userName: postgres
serverName: localhost
portNumber: 5432
Connected to database
org.postgresql.jdbc.PgConnection@387c703b
SQLState: 23505
Error Code: 0
Message: ERROR: duplicate key value violates unique constraint "pk_debito"
        Detail: Key (numero_debito)=(3000) already exists.
Um debito em IB2 inserido em 58 milisegundos
SQLState: 23505
Error Code: 0
Message: ERROR: duplicate key value violates unique constraint "pk_debito"
        Detail: Key (numero_debito)=(3001) already exists.
Um debito em IB2 inserido em 22 milisegundos
Releasing all open resources ...
mury@ubuntu:~/JDBCTutorial$

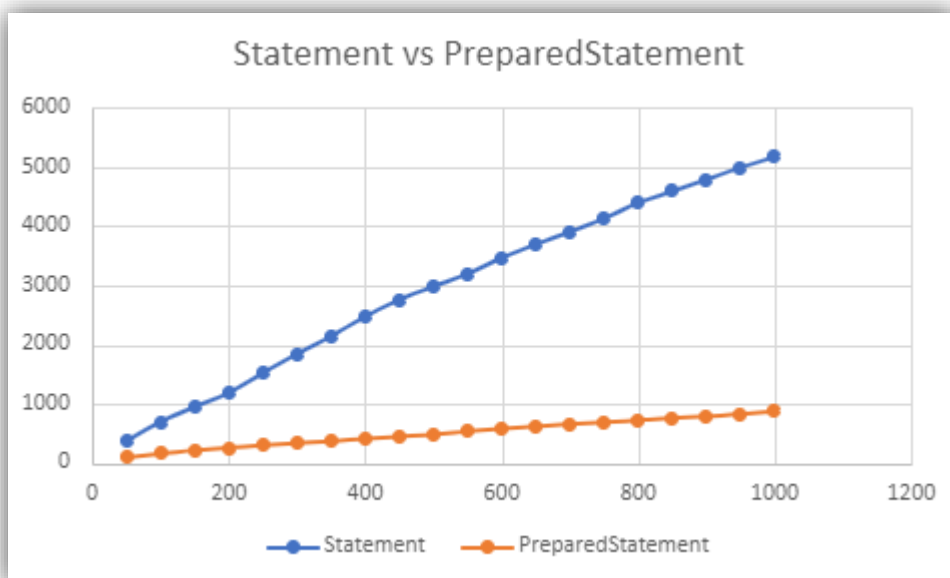
```

4) Agora você fará um loop para inserir milhares de linhas na tabela débito, uma vez para cada método copiado de insertMyData1 e insertMyData2. Todas as linhas serão quase que idênticas às linhas inseridas por insertMyData1 e insertMyData2, apenas com a diferença de que os campos numero\_debito e valor\_debito, da tabela debito, serão um único número sequencial gerado pelo laço. Para esse propósito, crie uma cópia dos métodos anteriores e renomeie estas cópias, como por exemplo, insertMyData1000 e insertMyData2000. Feito isto, introduza as seguintes alterações nos códigos:

```

dbms: postgresql
driver: org.postgresql.Driver
dbName: IB2
userName: postgres
serverName: localhost
portNumber: 5432
Connected to database
org.postgresql.jdbc.PgConnection@387c703b
50      371
100     694
150     947
200     1183
250     1521
300     1841
350     2144
400     2474
450     2761
500     2983
550     3198
600     3467
650     3697
700     3906
750     4133
800     4397
850     4593
900     4792
950     4994
1000    5174
Prepare Statement:
50      88
100     155
150     214
200     254

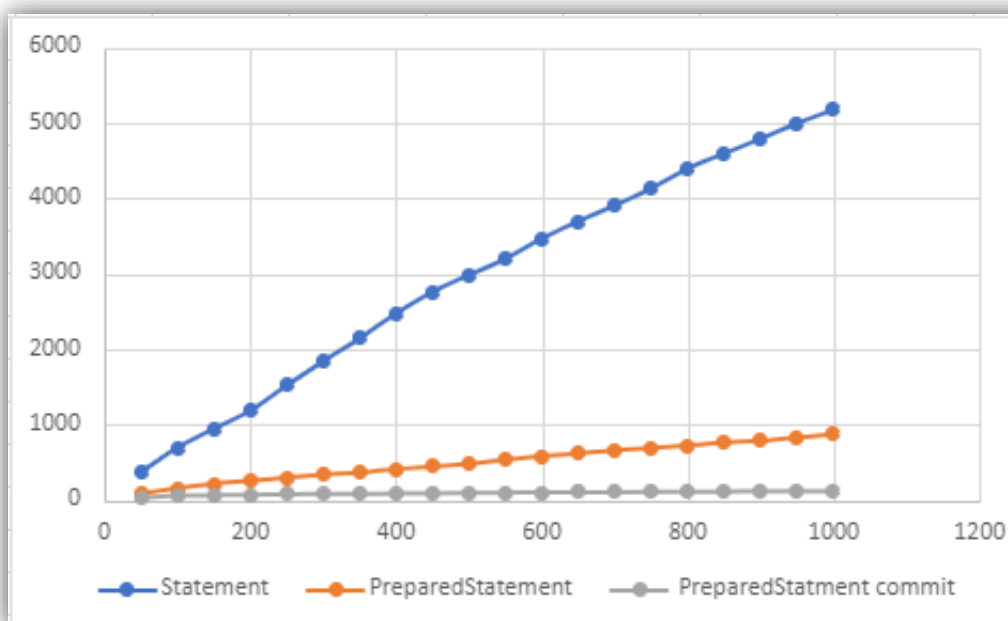
```



A partir do gráfico é possível concluir que o uso de PreparedStatement é mais eficiente que Statement.

5) Agora façamos uso do processamento em BATCH (lote) para executar o método insertMyData2000. Faça uma cópia do método insertMyData2000 e chame-o de insertMyData3000. Antes do comando for, o laço introduzido no passo anterior, acrescente este código abaixo. Porém, lembre-se que o código que cria o stmt deve ser removido do laço. Na sequência existe outro bloco de comandos que deve ser acrescentado após o comando for.

```
mury@ubuntu:~/JDBCTutorial$ ./comp MyQueries properties/postgres-properties.xml
Set the following properties:
dbms: postgresql
driver: org.postgresql.Driver
dbName: IB2
userName: postgres
serverName: localhost
portNumber: 5432
Connected to database
org.postgresql.jdbc.PgConnection@387c703b
50      27
100     59
150     62
200     65
250     76
300     79
350     80
400     86
450     88
500     91
550     95
600    101
650    102
700    104
750    105
800    106
850    110
900    117
950    117
1000   118
Releasing all open resources ...
mury@ubuntu:~/JDBCTutorial$
```



6) Como exercício, deixo uma pergunta para vocês responderem: O que acontecerá se utilizarmos o recurso de desabilitar o auto-commit para o caso de usarmos a classe Statement em vez de PreparedStatement?

```
mury@ubuntu:~/JDBCTutorial$ ./comp MyQueries properties/postgres-properties.xml
Set the following properties:
dbms: postgresql
driver: org.postgresql.Driver
dbName: IB2
userName: postgres
serverName: localhost
portNumber: 5432
Connected to database
org.postgresql.jdbc.PgConnection@387c703b
Debitos da Instituicao Bancaria atualizados.
Um debito em IB2 inserido em 11 milisegundos
Releasing all open resources ...
mury@ubuntu:~/JDBCTutorial$ ./comp MyQueries properties/postgres-properties.xml
Set the following properties:
dbms: postgresql
driver: org.postgresql.Driver
dbName: IB2
userName: postgres
serverName: localhost
portNumber: 5432
Connected to database
org.postgresql.jdbc.PgConnection@387c703b
Debitos da Instituicao Bancaria atualizados.
Um debito em IB2 inserido em 8 milisegundos
Releasing all open resources ...
mury@ubuntu:~/JDBCTutorial$
```

Nesse caso, o método leva mais tempo para ser executado.