

Carlos Augusto Dantas Marquez 11811BSI228

Matheus Costa Monteiro 12111BSI281

2

```
15
16      select dep.nome_cliente,
17      sum(dep.saldo_deposito), sum(emp.valor_emprestimo)
18      FROM deposito AS dep left Join emprestimo AS emp
19      on dep.nome_cliente = emp.nome_cliente
20      WHERE dep.saldo_deposito > 0 and emp.valor_emprestimo > 0
21      GROUP BY dep.nome_cliente, dep.nome_agencia, dep.numero_conta
```

Data Output Messages Notifications

	nome_cliente character varying (80)	sum double precision	sum double precision
47	Everardo Monfort Leitão	7392.56	4370.67
48	Thiago Leôncio Guimarães	11087.25	6528.7
49	Frederico de Oliveira Guimarães	209.52	1992.46
50	Marcos Andrade	14625.73	6308.01
51	Marta Avelar Santos	64291.76	17770.1
52	Germano Luiz de Paula	2585.51	1415.98
53	Maria Raquel Teixeira de Castro	2954.16	582.87

3

```
carlos@carlos-dell:~/bd2/JDBCTutorial$ ./comp MyQueries properties/postgres-properties2.xml
Set the following properties:
dbms: postgresql
driver: org.postgresql.Driver
dbName: IB2
userName: postgres
serverName: localhost
portNumber: 5432
Connected to database
Nome Cliente -- Soma Depósitos -- Soma Empréstimos
Gilmar Negreiros Carvalho - 572.77 - 410.1
Bruno Miranda Pacheco de Castro - 13617.04 - 4397.48
Manuel Oliveira - 2196.07 - 1396.84
Bruno Miranda Pacheco de Castro - 11779.28 - 4397.48
Rafael Mesquita Gondim - 9506.72 - 2607.32
Josenaldo Teixeira Soares - 3090.27 - 1380.92
Marta Avelar Santos - 24925.52 - 10662.06
Rafael Mesquita Gondim - 4880.28 - 5214.64
Everardo Monfort Leitão - 11089.76 - 4370.67
Andrade de Freitas - 6344.4 - 3147.63
```

4

1.2

Connection

```
select dep.nome_cliente, dep.nome_agencia, dep.numero_conta,
sum(dep.saldo_deposito) as soma_deposito, sum(emp.valor_emprestimo) as soma_emprestimo
FROM deposito AS dep left Join emprestimo AS emp
ON dep.nome_cliente = emp.nome_cliente AND dep.nome_agencia = emp.nome_agencia AND dep.numero_conta = emp.numero_conta
WHERE dep.saldo_deposito > 0 and emp.valor_emprestimo > 0
GROUP BY dep.nome_cliente, dep.nome_agencia, dep.numero_conta
```

a Output Messages Notifications

nome_cliente character varying (80)	nome_agencia character varying (50)	numero_conta integer	soma_deposito double precision	soma_emprestimo double precision
Alexandre Márcio de Souza	Gameleira	24067	4206.56	1030.12
Andrade de Freitas	Central	42436	13990.68	6295.26
André Cabral da Silva	Cidade Jardim	50073	6308.32	3298.32
André Cabral da Silva	Gameleira	93125	1752.23	771.98
Bruno Miranda Pacheco de Ca...	Gameleira	77069	3653.51	1522.88
Bruno Miranda Pacheco de Ca...	Pampulha	54505	3404.26	1469.77
Bruno Miranda Pacheco de Ca...	UFU	54194		

✓ Successfully run. Total query runtime: 89 msec. 31 rows affected. ✕

```
public static void cursorHoldabilitySupport(Connection conn)
    throws SQLException {
    DatabaseMetaData dbMetaData = conn.getMetaData();
    System.out.println("ResultSet.HOLD_CURSORS_OVER_COMMIT = " +
        ResultSet.HOLD_CURSORS_OVER_COMMIT);
    System.out.println("ResultSet.CLOSE_CURSORS_AT_COMMIT = " +
        ResultSet.CLOSE_CURSORS_AT_COMMIT);
    System.out.println("Default cursor holdability: " +
        dbMetaData.getResultSetHoldability());
    System.out.println("Supports HOLD_CURSORS_OVER_COMMIT? " +
        dbMetaData.supportsResultSetHoldability(
            ResultSet.HOLD_CURSORS_OVER_COMMIT));
    System.out.println("Supports ResultSet CONCUR_READ_ONLY? " +
        dbMetaData.supportsResultSetConcurrency(
            ResultSet.TYPE_FORWARD_ONLY, ResultSet.CONCUR_READ_ONLY));
    System.out.println("Supports ResultSet CONCUR_UPDATABLE? " +
        dbMetaData.supportsResultSetConcurrency(
            ResultSet.TYPE_FORWARD_ONLY, ResultSet.CONCUR_UPDATABLE));
}
```

getMyQueries3 + cursorHoldabilitySupport

The screenshot shows an IDE with a project named 'postgres-properties2.xml'. The 'MyQueries3.java' file is open, displaying a SQL query that is identical to the one in the first image. The 'Statement' window shows the query being executed, and the 'Terminal' window displays the output of the 'getMyQueries3' method, including the query text and the results of the 'ResultSet' object.

Terminal Output:

```

Maria Raquel Teixeira de Castro - Cidade Jardim-57745-3181.8682 - 582.87
Marta Avelar Santos - Glória-82296-4880.9355 - 1455.5
Marta Avelar Santos - Pampulha-4552-50629.76 - 14131.35
Rafael Mesquita Gondim - Central-50906-1281.0735 - 642.14
Rafael Mesquita Gondim - Pampulha-7383-4991.028 - 287.22
Thiago Leôncio Guimarães - Glória-30251-3888.5376 - 1528.58
ResultSet.HOLD_CURSORS_OVER_COMMIT = 1
ResultSet.CLOSE_CURSORS_AT_COMMIT = 2
Default cursor holdability: 1
Supports HOLD_CURSORS_OVER_COMMIT? true
Supports ResultSet CONCUR_READ_ONLY? true
Supports ResultSet CONCUR_UPDATABLE? true
Releasing all open resources ...

```

modify prices que funciona com ou sem taxa modificada

```

public static void modifyPrices(Connection conn, float rate, boolean custom) throws SQLException {
    Statement stmt = null;
    rate = custom ? rate : 0.5f;
    try {
        stmt = conn.createStatement();
        stmt = conn.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
            ResultSet.CONCUR_UPDATABLE);
        ResultSet uprs = stmt.executeQuery( sql: "SELECT * FROM DEPOSITO");
        while (uprs.next()) {
            float ratee = uprs.getFloat( columnLabel: "SALDO_DEPOSITO");
            uprs.updateFloat( columnLabel: "SALDO_DEPOSITO", x: ratee + rate);
            uprs.updateRow();
        }
    } catch (SQLException e) {
        JDBCUtilities.printSQLException(e);
    } finally {
        if (stmt != null) { stmt.close(); }
    }
}

```

inserção das linhas das tabelas

```

135 @ public static void insertRow(Connection con) throws SQLException {
136     Statement stmt = null;
137     try {
138         stmt = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
139             ResultSet.CONCUR_UPDATABLE);
140         ResultSet uprs = stmt.executeQuery( sql: "SELECT * FROM debito");
141         uprs.moveToInsertRow();
142         uprs.updateInt( columnIndex: 1, x: 2000);
143         uprs.updateFloat( columnIndex: 2, x: 150);
144         uprs.updateInt( columnIndex: 3, x: 1);
145         uprs.updateDate( columnIndex: 4, Date.valueOf( s: "2014-01-23"));
146         uprs.updateInt( columnIndex: 5, x: 46248);
147         uprs.updateString( columnIndex: 6, x: "UFU");
148         uprs.updateString( columnIndex: 7, x: "Carla Soares Sousa");
149         uprs.insertRow();
150
151         uprs.moveToInsertRow();
152         uprs.updateInt( columnIndex: 1, x: 2001);
153         uprs.updateFloat( columnIndex: 2, x: 200);
154         uprs.updateInt( columnIndex: 3, x: 2);
155         uprs.updateDate( columnIndex: 4, Date.valueOf( s: "2014-01-23"));
156         uprs.updateInt( columnIndex: 5, x: 26892);
157         uprs.updateString( columnIndex: 6, x: "Glória");

```

menu para rodar:

```
public static void menu(Connection con){
try{
    //para rodar só deixar descomentado a função que quiser, está na ordem que o sr pediu

    Scanner in = new Scanner(System.in);
    float rate = 0;

    MyQueries.getMyData3(con);

    MyQueries.cursorHoldabilitySupport(con);

    System.out.println("taxa: \n" );
    rate = in.nextFloat();
    MyQueries.modifyPrices(con,rate,true);

    insertRow(con);
}
catch (Exception e) {
    System.out.println(e.getMessage());
}
```