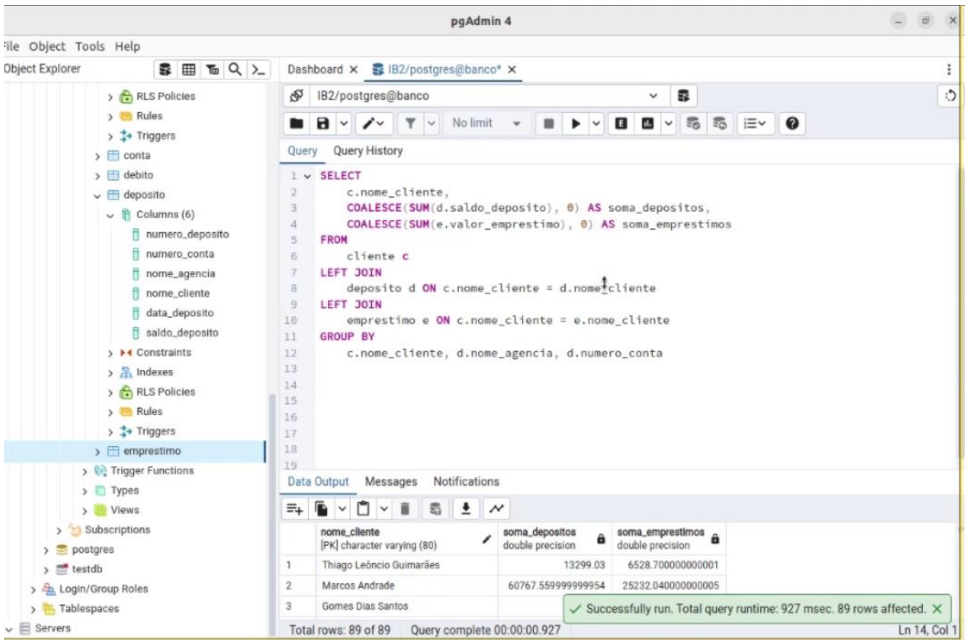
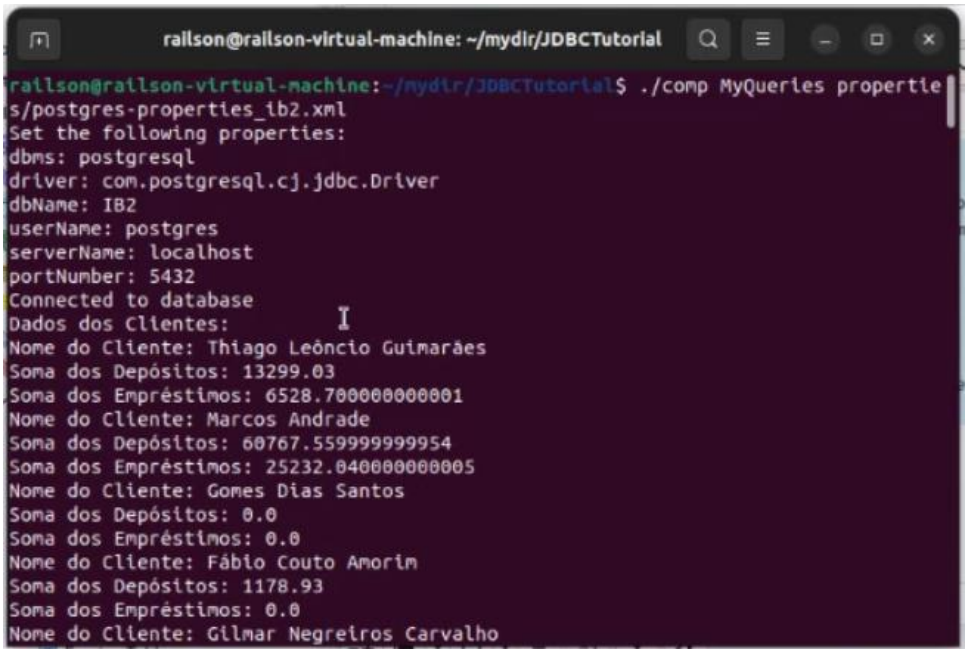
T07-Recuperando e modificando dados

Railson Da Silva Martins - 11811BSI208
Wothon Mateus de Araujo 12111BSI262

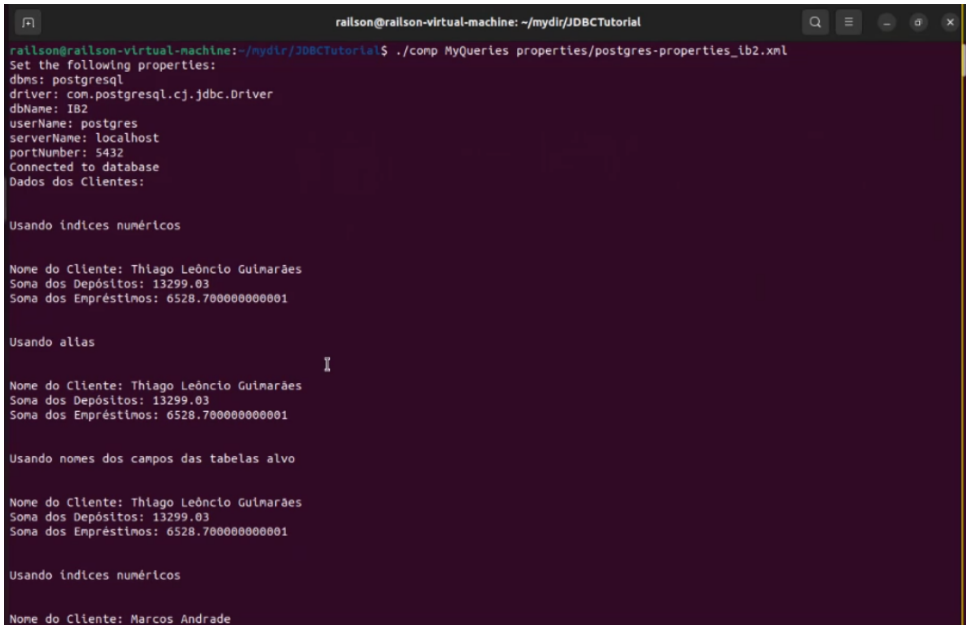
Print referente ao passo 1: Executado a função criada sql



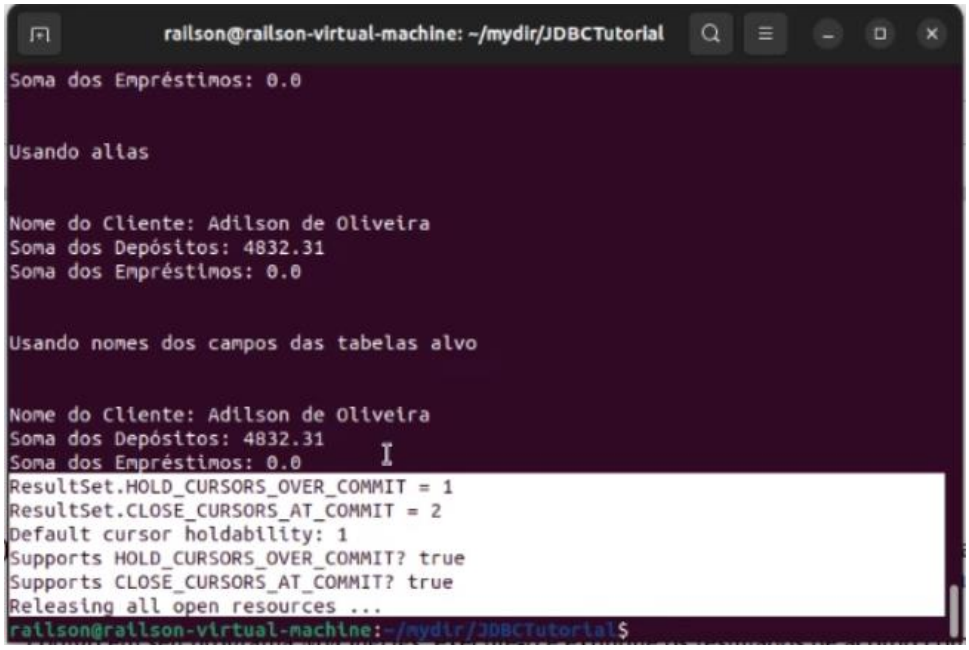
Print referente ao passo 2: Modificamos o arquivo MyQueries o print e da execução com o comando comp:



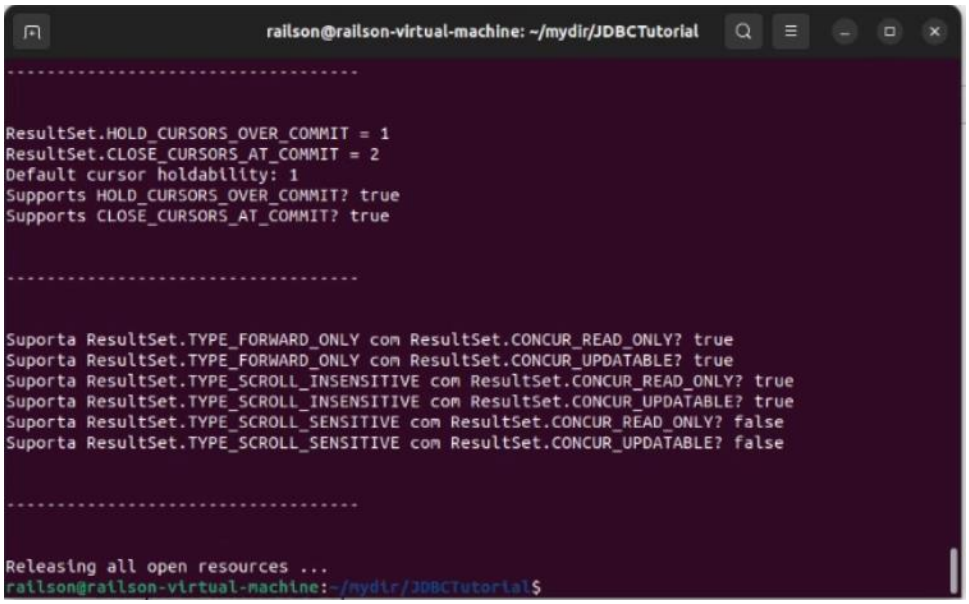
Print referente ao passo 3: Modifique o programa do item 2 para retornar os dados da consulta solicitada no item 1. Código fonte em um arquivo separado print da e execução.



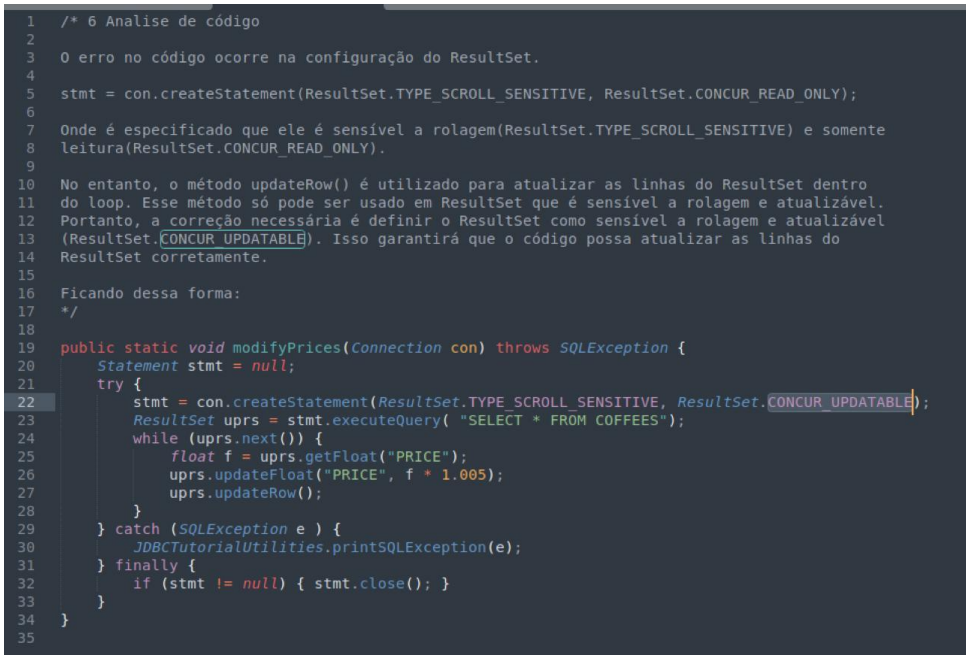
Print referente ao passo 4: Foi modificado novamente o arquivo MyQueries print da execução do novo MyQueries.



Print referente ao passo 5:

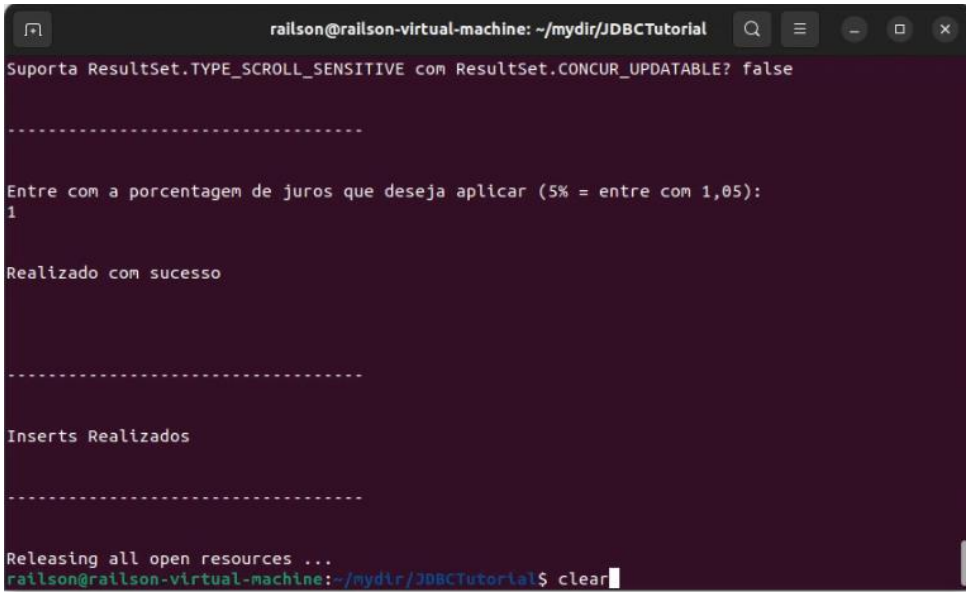


Print referente ao passo 6: Print da análise feita e com o código ficou após a correção

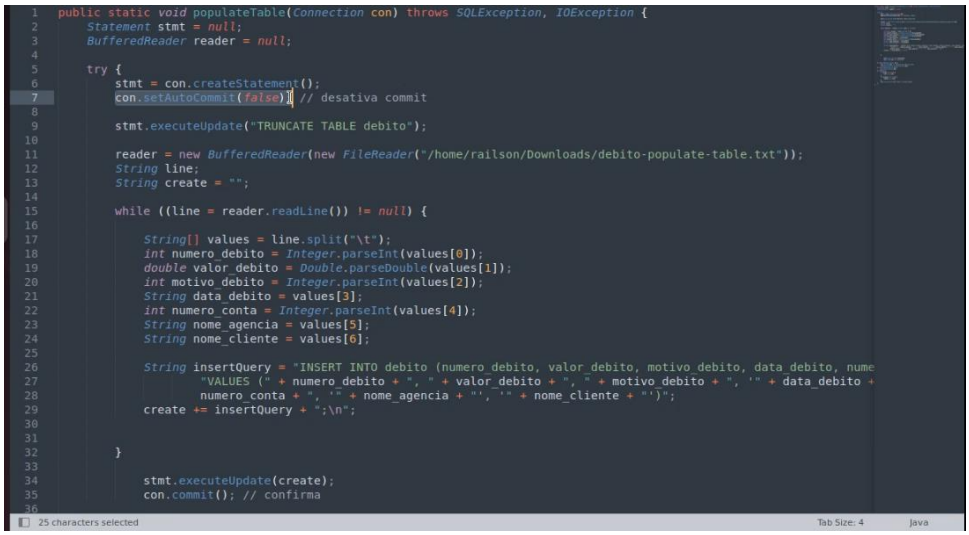


```
/* 6 Analise de código

O erro no código ocorre na configuração do ResultSet.

stmt = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_READ_ONLY);

Onde é especificado que ele é sensível a rolagem(ResultSet.TYPE_SCROLL_SENSITIVE) e somente
leitura(ResultSet.CONCUR_READ_ONLY).

No entanto, o método updateRow() é utilizado para atualizar as linhas do ResultSet dentro
do loop. Esse método só pode ser usado em ResultSet que é sensível a rolagem e atualizável.
Portanto, a correção necessária é definir o ResultSet como sensível a rolagem e atualizável
(ResultSet.CONCUR_UPDATABLE). Isso garantirá que o código possa atualizar as linhas do
ResultSet corretamente.

Ficando dessa forma:
*/

public static void modifyPrices(Connection con) throws SQLException {
    Statement stmt = null;
    try {
        stmt = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE);
        ResultSet uprs = stmt.executeQuery( "SELECT * FROM COFFEES");
        while (uprs.next()) {
            float f = uprs.getFloat("PRICE");
            uprs.updateFloat("PRICE", f * 1.005);
            uprs.updateRow();
        }
    } catch (SQLException e ) {
        JDBCTutorialUtilities.printSQLException(e);
    } finally {
        if (stmt != null) { stmt.close(); }
    }
}
```

Print referente ao passo 7 e 8: Print com a opção de escolhe o juro que deve ser aplicada aos débitos. Código entregue em um arquivo separado

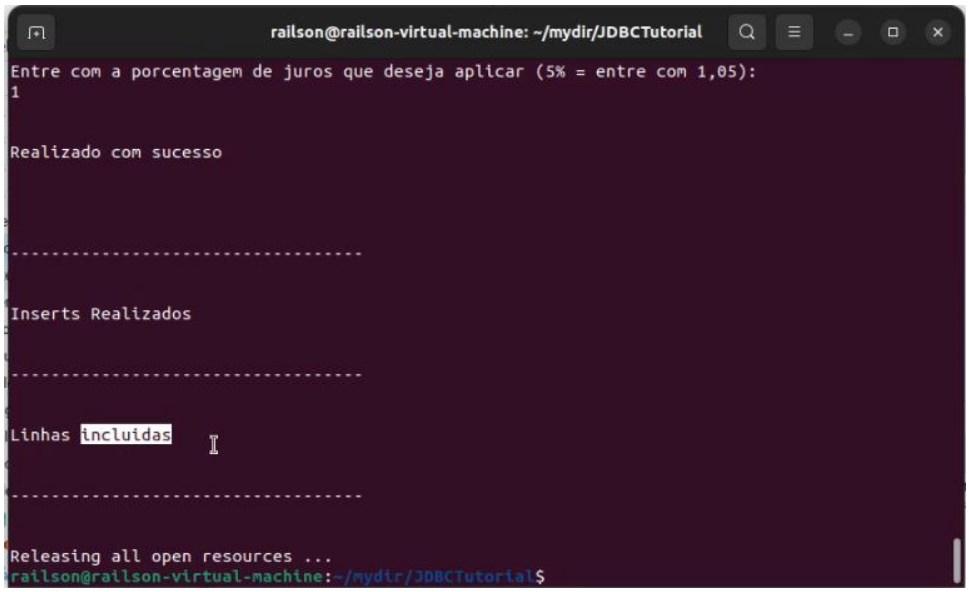


Print referente ao passo 9: No print anterior a saída “Inserts Realizado” e reverente ao código solicitado no passo 9 sendo executado. Abaixo o print do código.



```java
public static void populateTable(Connection con) throws SQLException, IOException {
    Statement stmt = null;
    BufferedReader reader = null;

    try {
        stmt = con.createStatement();
        con.setAutoCommit(false); // desativa commit

        stmt.executeUpdate("TRUNCATE TABLE debito");

        reader = new BufferedReader(new FileReader("/home/railson/Downloads/debito-populate-table.txt"));
        String line;
        String create = "";

        while ((line = reader.readLine()) != null) {

            String[] values = line.split("\t");
            int numero_debito = Integer.parseInt(values[0]);
            double valor_debito = Double.parseDouble(values[1]);
            int motivo_debito = Integer.parseInt(values[2]);
            String data_debito = values[3];
            int numero_conta = Integer.parseInt(values[4]);
            String nome_agencia = values[5];
            String nome_cliente = values[6];

            String insertQuery = "INSERT INTO debito (numero_debito, valor_debito, motivo_debito, data_debito, nume
                    "VALUES (" + numero_debito + ", " + valor_debito + ", " + motivo_debito + ", '" + data_debito +
                    numero_conta + ", '" + nome_agencia + "', '" + nome_cliente + "')";
            create += insertQuery + ";\n";


        }

            stmt.executeUpdate(create);
            con.commit(); // confirma
```

Print referente ao passo 10: Executado o código feito no passo 10 “Linhas incluídas”



Print referente ao passo 10.1: Indicado que realmente a linhas foram incluídas com sucesso.