package Server;

import java.io.\*;

import java.net.\*;

import java.sql.\*;

import java.util.concurrent.\*;

public class Server implements Runnable {

private final Socket SV;

public Server(Socket clientSocket) {

this.SV = clientSocket;

}

@Override

public void run() {

DataInputStream DIS = null;

DataOutputStream DOS = null;

Connection connection = null;

try {

DIS = new DataInputStream(SV.getInputStream());

int repID = DIS.readInt();

int Sold = DIS.readInt();

if (Sold < 0) {

throw new IllegalArgumentException("Number of laptops sold cannot be negative.");

}

double Price = 90.0;

double sales = Sold \* Price;

String chargeCode = (sales > 20000) ? "CR1" :

(sales > 10000) ? "CR2" : "CR3";

String databese = "jdbc:derby://localhost:1527/MEC";

String username = "MEC\_2022";

String password = "123456";

connection = DriverManager.getConnection(databese, username , password);

PreparedStatement statement = connection.prepareStatement("SELECT ChargeRate FROM ChargeRates WHERE ChargeCode = ?");

statement.setString(1, chargeCode);

ResultSet resultSet = statement.executeQuery();

double Rate = 0.0;

if (resultSet.next()) {

Rate = resultSet.getDouble("ChargeRate");

}

double Value = sales \* (Rate / 100);

DOS = new DataOutputStream(SV.getOutputStream());

DOS.writeDouble(sales);

DOS.writeDouble(Rate);

DOS.writeDouble(Value);

System.out.printf("Processed SalesRep ID: %d | Profit: %.2f | Rate: %.2f%% | Commission: %.2f%n",

repID, sales, Rate, Value);

} catch (IOException | SQLException | IllegalArgumentException e) {

System.out.println("Error: " + e.getMessage());

} finally {

try {

if (DIS != null) DIS.close();

if (DOS != null) DOS.close();

if (SV != null) SV.close();

if (connection != null) connection.close();

} catch (IOException | SQLException e) {

System.out.println("Error closing resources: " + e.getMessage());

}

}

}

public static void main(String[] args) {

ExecutorService executor = Executors.newFixedThreadPool(10);

try (ServerSocket serverSocket = new ServerSocket(7195)) {

System.out.println("Server is running on port 7195...");

while (true) {

Socket clientSocket = serverSocket.accept();

System.out.println("Client connected.");

executor.execute(new Server(clientSocket));

}

} catch (IOException e) {

System.out.println("Server is runn");

} finally {

executor.shutdown();

}

{

}