Q1-

Answer:-

import java.io.OutputStream;

import java.net.HttpURLConnection;

import java.net.URL;

public class App {

public static void main(String[] args) {

sendStartupRequest();

}

public static void sendStartupRequest() {

try {

// Define the target URL

URL url = new URL("https://bfhldevapigw.healthrx.co.in/hiring/generateWebhook/JAVA");

HttpURLConnection conn = (HttpURLConnection) url.openConnection();

// Set up the request

conn.setRequestMethod("POST");

conn.setRequestProperty("Content-Type", "application/json");

conn.setDoOutput(true);

// Define the JSON body

String jsonInputString = """

{

"name": "John Doe",

"regNo": "REG12347",

"email": "john@example.com"

}

""";

// Write the JSON data to the output stream

try (OutputStream os = conn.getOutputStream()) {

byte[] input = jsonInputString.getBytes("utf-8");

os.write(input, 0, input.length);

}

// Check the response

int responseCode = conn.getResponseCode();

System.out.println("POST Response Code: " + responseCode);

if (responseCode == HttpURLConnection.HTTP\_OK || responseCode == HttpURLConnection.HTTP\_CREATED) {

System.out.println("POST request successful.");

} else {

System.out.println("POST request failed.");

}

} catch (Exception e) {

e.printStackTrace();

}

}

}

Q2-

Answers

import java.sql.\*;

import java.time.LocalDate;

import java.time.Period;

import java.util.HashMap;

import java.util.Map;

public class YoungerEmployeesCounter {

public static void main(String[] args) {

String url = "jdbc:mysql://localhost:3306/your\_database\_name";

String username = "your\_username";

String password = "your\_password";

try (Connection conn = DriverManager.getConnection(url, username, password)) {

// Step 1: Fetch all employees with their DOB and department

Map<Integer, EmployeeInfo> employees = new HashMap<>();

Map<Integer, DepartmentEmployees> departmentEmployees = new HashMap<>();

String employeeQuery = "SELECT e.EMP\_ID, e.FIRST\_NAME, e.LAST\_NAME, e.DOB, e.DEPARTMENT, d.DEPARTMENT\_NAME " +

"FROM EMPLOYEE e JOIN DEPARTMENT d ON e.DEPARTMENT = d.DEPARTMENT\_ID " +

"ORDER BY e.EMP\_ID DESC";

try (Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery(employeeQuery)) {

while (rs.next()) {

int empId = rs.getInt("EMP\_ID");

String firstName = rs.getString("FIRST\_NAME");

String lastName = rs.getString("LAST\_NAME");

Date dob = rs.getDate("DOB");

int departmentId = rs.getInt("DEPARTMENT");

String departmentName = rs.getString("DEPARTMENT\_NAME");

EmployeeInfo empInfo = new EmployeeInfo(empId, firstName, lastName, dob, departmentId, departmentName);

employees.put(empId, empInfo);

// Group employees by department

departmentEmployees.computeIfAbsent(departmentId, k -> new DepartmentEmployees())

.addEmployee(empInfo);

}

}

// Step 2: Calculate younger employees count for each employee

String outputQuery = "SELECT e.EMP\_ID, e.FIRST\_NAME, e.LAST\_NAME, d.DEPARTMENT\_NAME, " +

"(SELECT COUNT(\*) FROM EMPLOYEE e2 WHERE e2.DEPARTMENT = e.DEPARTMENT AND e2.DOB > e.DOB) AS YOUNGER\_EMPLOYEES\_COUNT " +

"FROM EMPLOYEE e JOIN DEPARTMENT d ON e.DEPARTMENT = d.DEPARTMENT\_ID " +

"ORDER BY e.EMP\_ID DESC";

// Alternatively, we can calculate in Java (commented out the SQL approach above)

System.out.println("EMP\_ID | FIRST\_NAME | LAST\_NAME | DEPARTMENT\_NAME | YOUNGER\_EMPLOYEES\_COUNT");

System.out.println("------------");

for (Map.Entry<Integer, EmployeeInfo> entry : employees.entrySet()) {

EmployeeInfo emp = entry.getValue();

DepartmentEmployees deptEmps = departmentEmployees.get(emp.departmentId);

int youngerCount = 0;

for (EmployeeInfo otherEmp : deptEmps.employees) {

if (otherEmp.dob.after(emp.dob)) {

youngerCount++;

}

}

System.out.printf("%-6d | %-10s | %-9s | %-15s | %d%n",

emp.empId, emp.firstName, emp.lastName, emp.departmentName, youngerCount);

}

} catch (SQLException e) {

e.printStackTrace();

}

}

static class EmployeeInfo {

int empId;

String firstName;

String lastName;

Date dob;

int departmentId;

String departmentName;

public EmployeeInfo(int empId, String firstName, String lastName, Date dob, int departmentId, String departmentName) {

this.empId = empId;

this.firstName = firstName;

this.lastName = lastName;

this.dob = dob;

this.departmentId = departmentId;

this.departmentName = departmentName;

}

}

static class DepartmentEmployees {

List<EmployeeInfo> employees = new ArrayList<>();

public void addEmployee(EmployeeInfo emp) {

employees.add(emp);

}

}

}