SOLUTION TO HOMEWORK 4 : CS 480 SPRING 2014
BY
KANISHKA GARG | UIN - 675005799

This homework has two schemas :

1. employee(eid:integer, name:string, salary:integer)
2. worksfor(eid:integer, mid:integer)

Programming language used : Java

My program is named as homework4.java

Homework4.java

```
import java.io.*;
import java.sql.*;
import java.util.*;

public class homework4 {

    private static final String file_path = "transfile.txt";

    public static void main(String[] args) {
        homework4 h4 = new homework4();
        Connection conn = null;
        conn = h4.conn_to_database();
        h4.create_table(conn);
        h4.read_file(conn, file_path);
        h4.delete_table(conn);
        h4.close_database(conn);
    }

    public Connection conn_to_database() {
        String url = "jdbc:mysql://localhost:3306/";
        String dbName = "hw4";
        String driver = "com.mysql.jdbc.Driver";
        String userName = "root";
        String password = "root";
        Connection conn = null;
        try {
            Class.forName(driver).newInstance();
            conn = DriverManager.getConnection(url + dbName, userName, password);
            System.out.println("Connection to the database is successful!");

        } catch (Exception ex) {
            ex.printStackTrace();
        }
        return conn;
    }

    public void close_database(Connection conn) {
```

```
		try {
			if (conn != null) {
				conn.close();
				System.out.println("Connection to the database is closed!");
			}
		} catch (SQLException e) {
			e.printStackTrace();
		}
	}

	private void create_table(Connection conn) {
		Statement stmt = null;
		String sqltable1 = "CREATE TABLE `hw4`.`employee`(`eid` INTEGER NOT NULL,
`name` VARCHAR(20) NOT NULL,`salary` INT(6),PRIMARY KEY(`eid`))";
		String sqltable2 = "CREATE TABLE `hw4`.`worksfor`(`eid` INTEGER NOT
NULL,`mid` INTEGER   NOT NULL,   PRIMARY KEY (`eid`, `mid`))";
		try {
			stmt = conn.createStatement();
			stmt.execute(sqltable1);
			stmt.execute(sqltable2);
			stmt.close();
		} catch (SQLException e) {
			e.printStackTrace();
		}
	}

	private void delete_table(Connection conn) {
		Statement stmt = null;
		String sqltable1 = "drop table `hw4`.`employee`";
		String sqltable2 = "drop table `hw4`.`worksfor`";
		try {
			stmt = conn.createStatement();
			stmt.execute(sqltable1);
			stmt.execute(sqltable2);
			stmt.close();
		} catch (SQLException e) {
			e.printStackTrace();
		}
	}

	private void read_file(Connection conn, String file_path_local) {
		BufferedReader reader = null;

		try {
			System.out.println("transfile.txt is getting read by the program.");
			reader = new BufferedReader(new FileReader(file_path_local));
			String line = null;
			while ((line = reader.readLine()) != null) {
				exec_file(conn, line);
			}

		} catch (IOException e) {
			System.out.println("Error while reading transfile.txt");
```

```
        } finally {
            if (reader != null) {
                try {
                    reader.close();
                    System.out.println("transfile.txt work over");
                } catch (IOException e) {

                }
            }
        }
    }

    private void exec_file(Connection conn, String line) {
        String[] tokens = line.split(" ");// Tokenize the line in the transfile. This split() will split on
the basis of delimiter " "[space] and each token will be stored as a string in the array.
        int code = Integer.parseInt(tokens[0]);// convert string token into int token because the first
token will be the code.

        try {
            Statement stmt = conn.createStatement();
            String query1 = null;

            switch (code) {
            case 1:
                try {
                    // delete the employee from employee table
                query1 = "delete from `hw4`.`employee` where    eid = " + tokens[1];
                    int res = stmt.executeUpdate(query1);

                    if (res == 0) {
            System.out.println("Error in deleting the employee from employee table");
                        break;
                    }

                    // delete all the corresponding tuples from worksfor table
                    try {
String query2 = "delete from `hw4`.`worksfor` where eid = " + tokens[1] + " or mid = "+ tokens[1];
stmt.executeUpdate(query2);
                    } catch (Exception e) {
                                System.out.println("Error   in   creating   SQL   statement   for
deleting all the corresponding tuples from worksfor table");
                                e.printStackTrace();
                    }

                    System.out.println("done");

                } catch (SQLException e) {
        System.out.println("Error in creating SQL statement for transaction code = 1");
                    e.printStackTrace();
                }
                break;

            case 2:
```

```
                        try {
                              // insert the eid,name,salary in the employee table
query1 = "insert into `hw4`.`employee` values(" + tokens[1] + ",'" + tokens[2] + "'," + tokens[3]+ ");";
                              stmt.executeUpdate(query1);

                              // insert all the distinct mid in the worksfor table
                  String query2 = "insert into `hw4`.`worksfor` values(" + tokens[1] + ",";

for (int i = 4; i < tokens.length; i++) {// Transaction code 4 data can have more than 3 arguments. This
loop take care of all the remaining arguments.
                                    try {
                                          stmt.executeUpdate(query2 + tokens[i] + ");");
                                    } catch (SQLException e) {
                                                      System.out.println("Error in creating SQL statement for
inserting all the distinct mid in the worksfor table");
                                                      e.printStackTrace();
                                    }
                              }

                              System.out.println("done");

                        } catch (Exception e) {
            System.out.println("Error in creating SQL statement for transaction code = 2");
                        }
                        break;

                  case 3:
query1 = "select avg(salary) from `hw4`.`employee`";// Retreiving the average salary from the
employee table

                        try {
                              ResultSet rs = stmt.executeQuery(query1);
                              String avg = null;

                              if (rs.next() && (avg = rs.getString(1)) != null)
                                    System.out.println(avg);
                              else
                                    System.out.println("error");

                        } catch (Exception e) {
                              System.out.println("Error in creating SQL statement for transaction code =
3");
                        }
                        break;

                  case 4:
                        try {
                              query1 = "select eid,name from `hw4`.`employee` where eid in "
                                          + "(select eid from `hw4`.`worksfor` where mid = ?);";

                              PreparedStatement p = conn.prepareStatement(query1);
                              p.setString(1, tokens[1]);
                              ResultSet rs = p.executeQuery();
                              if (!rs.next()) {
                                    System.out.println("error");
                                    break;
                              }
                              Stack<String> emp_id = new Stack<String>();
                              ArrayList<String> emp_id_1 = new ArrayList<String>();
                              HashSet<String> name = new HashSet<String>();
                              rs.beforeFirst();

                              while (rs.next()) {
                                    emp_id.push(rs.getString(1));
                                    name.add(rs.getString(2));
                              }

                              while (!emp_id.isEmpty()) {
                                    String eid = emp_id.pop();
                                    emp_id_1.add(eid);
                                    p.setString(1, eid);
                                    rs = p.executeQuery();

                                    while (rs.next()) {
                                          if (!emp_id_1.contains(rs.getString(1)))
                                              emp_id.push(rs.getString(1));
                                          name.add(rs.getString(2));
                                    }
                              }

                              for (String x : name) {
                                    System.out.println("Employee Name :" + x);
                              }

                        } catch (Exception e) {
                              System.out.println("Error in creating SQL statement for transaction code =
4");
                        }
                        break;

                  case 5:
                        try {
                              query1 = "select avg(`hw4`.`employee`.salary) from `hw4`.`employee` where
eid in "+ "(select eid from `hw4`.`worksfor` where mid = " + tokens[1] + ");";
                              ResultSet rs = stmt.executeQuery(query1);
                              String avg = null;

                              if (rs.next() && (avg = rs.getString(1)) != null)
                                    System.out.println((int) Math.round(Double.parseDouble(avg)));
                              else
                                    System.out.println("error");

                        } catch (Exception e) {
                              System.out.println("Error in creating SQL statement for transaction code =
5");
```

```
                        e.printStackTrace();
                    }
                    break;

                case 6:
                    query1 = "select name from `hw4`.`employee` natural join `hw4`.`worksfor` group
by eid having count(mid) > 1;";

                    try {
                        ResultSet rs = stmt.executeQuery(query1);
                        if (!rs.next()) {
                            System.out.println("No employees with more than one manager");
                            break;
                        }
                        rs.beforeFirst();
                        while (rs.next()) {
                            System.out.println(rs.getString(1));
                        }

                    } catch (Exception e) {
                        System.out.println("error");
                    }
                    break;}

            } catch (SQLException e) {
                System.out.println("Error in creating SQL statement for transaction code = 6");
            }
    }}
```

READ ME INSTRUCTIONS:

Put the homework4.java and mysql-connector-java-5.1.12-bin.jar in the same directory.

Compile:

```
javac -cp mysql-connector-java-5.1.12-bin.jar homework4.java
```

Run:

```
java -cp mysql-connector-java-5.1.12-bin.jar:. homework4
```

SAMPLE Transfile.txt (which I used) :

```
2 100 Mary 50000 50
2 101 Ford 53000 100
2 50 Bell 55000 0 300
2 300 Vikas 44000 50 100
6
5 50
4 50
3
1 100
```