Jason Mejia 7/9/20

CSC336 Project 2

Family Relations – II

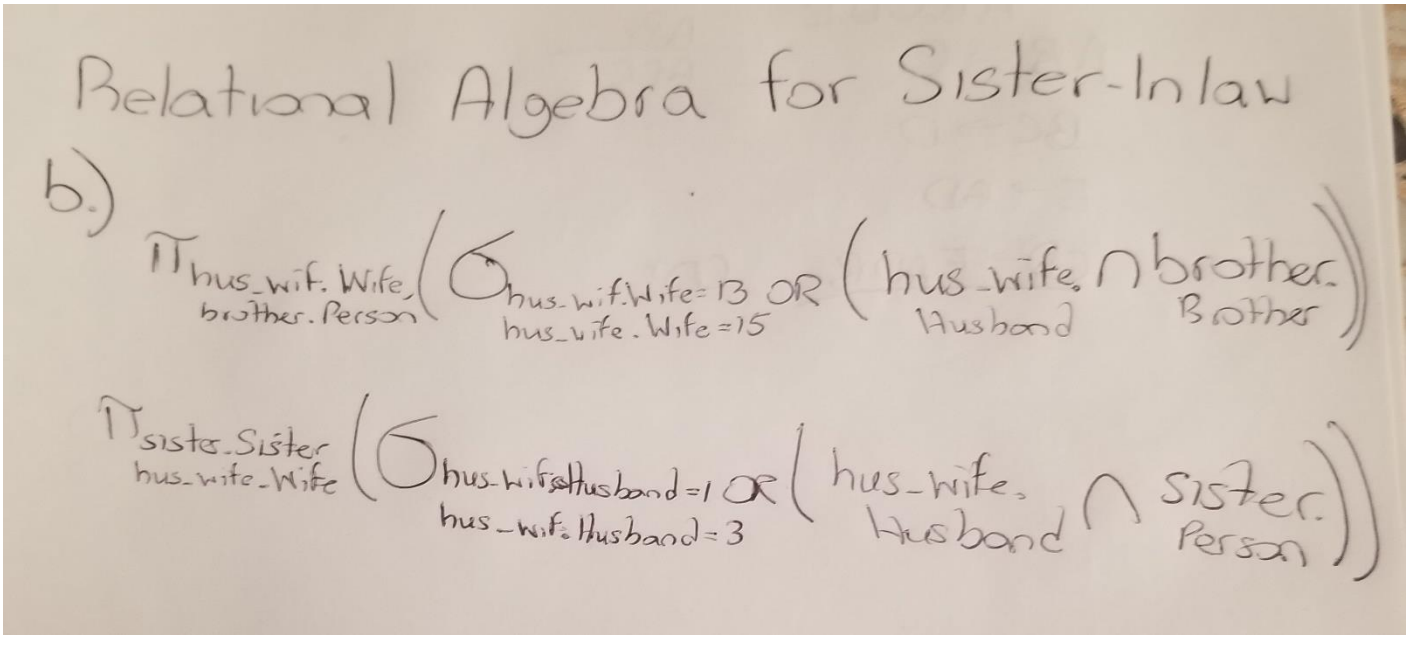
Problem:

In this project we need to store information in the database using a java application. Since I am using MySQL workbench, I have to find a way to interact with MySQL by using java. From the previous assignment creating the tables such Persons, Family, grandparents and now adding more tables, it needs to be done in java. The given tables are Brother, Sister, Brother/Sister, and Husband/Wife. These tables are needed to fine the tuples for the table Sister-in-law. And solving the relational algebra for the table Sister-in-law.

Solution:

I am using Eclipse as my java application for the project. Now I have to use JDBC which will be used to connect to the database I am using in MySQL. Once we are connected to the database we can input the data we need. For the tables such as Persons, Family and grandparents are being used again from the first assignment. Now we are adding the values for the tables Brother, Sister, Brother/Sister, and Husband/Wife. For Sister-in-law we used the tables Brother, Sister, and Husband/Wife relationships to find the tuples for this table. For certain relationships such as being a wife to a husband that has multiple siblings, we used the tables brother and husband/wife. If a sister has brothers that are married then I will use the tables sister and husband/wife for that relationship.

What couldn't achieve in this assignment was putting proper constructors for the classes and the proper functions required for the project.



Code:

Persons.java

```
package relations;
import java.sql.*;

public class Persons {

	public static void main(String[] args) {

		String url = "jdbc:mysql://localhost:3306/people";
		String username = "root";
		String password = "Something11";

		try {
			// Connecting to the database from MySQL workbench
			System.out.println("Connecting to a selected database...");
			Connection myConn = DriverManager.getConnection(url, username, password);
			System.out.println("Connected database successfully...");
			Statement myStmt = myConn.createStatement();

			//Here we started to create the first table Persons, which provide the number
			 of people to rest of the tables.
			String sql = "CREATE TABLE Persons " +
					"(ID INT PRIMARY KEY, " +
					"Name VARCHAR(20), " +
					"DOB VARCHAR(20) DEFAULT 'Unknown', " +
					"Gender VARCHAR(1) CHECK(Gender = 'M' OR Gender = 'F')) ";
			myStmt.executeUpdate(sql);
			System.out.println("Created table in given database...");

			//Inserting the values into the table
			String insert = "INSERT INTO Persons VALUES(1, 'Jason', '11/30/1996', 'M') ";
			myStmt.executeUpdate(insert);
			insert = "INSERT INTO Persons VALUES(2, 'Nancy', '12/01/1989', 'F') ";
			myStmt.executeUpdate(insert);
			insert = "INSERT INTO Persons VALUES(3, 'Alfredo', '03/03/1987', 'M') ";
			myStmt.executeUpdate(insert);
			insert = "INSERT INTO Persons VALUES(4, 'Raquel', '04/27/1961', 'F') ";
			myStmt.executeUpdate(insert);
			insert = "INSERT INTO Persons VALUES(5, 'Giselo', '05/07/1960', 'M') ";
			myStmt.executeUpdate(insert);
			insert = "INSERT INTO Persons VALUES(6, 'Evelyn', '08/16/2012', 'F') ";
			myStmt.executeUpdate(insert);
			insert = "INSERT INTO Persons VALUES(7, 'Julian', '11/29/2014', 'M') ";
			myStmt.executeUpdate(insert);
			insert = "INSERT INTO Persons VALUES(8, 'Chris', '12/02/1989', 'M') ";
			myStmt.executeUpdate(insert);
			insert = "INSERT INTO Persons VALUES(9, 'Irene', '02/16/1908', 'F') ";
			myStmt.executeUpdate(insert);
			insert = "INSERT INTO Persons VALUES(10, 'Hector', '6/28/1908', 'M') ";
			myStmt.executeUpdate(insert);
			insert = "INSERT INTO Persons VALUES(11, 'Bertha', '6/28/1943', 'F') ";
			myStmt.executeUpdate(insert);
			insert = "INSERT INTO Persons VALUES(12, 'Michael', '6/28/1943', 'M') ";
			myStmt.executeUpdate(insert);
			insert = "INSERT INTO Persons VALUES(13, 'Lisa', '8/15/1996', 'F') ";
			myStmt.executeUpdate(insert);
			insert = "INSERT INTO Persons VALUES(14, 'Oliver', '3/21/2021', 'M') ";
			myStmt.executeUpdate(insert);
```

```
                insert = "INSERT INTO Persons VALUES(15, 'Kim', '3/4/1987', 'F') ";
                myStmt.executeUpdate(insert);
                System.out.println("Inserted records into the table...");

                // This is where we would be calling the table and see all the information
                ResultSet myRs = myStmt.executeQuery("select * from Persons");
                while (myRs.next()) {
                        int id  = myRs.getInt("ID");
                        String name = myRs.getString("Name");
                        String dob = myRs.getString("DOB");
                        String gender = myRs.getString("Gender");

                        System.out.println("ID:" + id +
                                                ", Name:" + name +
                                                ", DOB:" + dob +
                                                ", Gender:" + gender);
                }

        }
        catch (Exception exc) {
                exc.printStackTrace();
        }

    }

}
```

Child.java

```
package relations;
import java.sql.*;

public class Child extends Persons {

	public static void main(String[] args) {
		String url = "jdbc:mysql://localhost:3306/people";
		String username = "root";
		String password = "Something11";

		try {
			//Connecting to the database for the class Child
			System.out.println("Connecting to a selected database...");
			Connection myConn = DriverManager.getConnection(url, username, password);
			System.out.println("Connected database successfully...");
			Statement myStmt = myConn.createStatement();
			ResultSet rs;

			//Creating the table Family to see the parents of a child
			String family = "CREATE TABLE Family " +
					"(Child INT, " +
					"Father INT, " +
					"Mother INT, " +
					"PRIMARY KEY (Child), " +
					"FOREIGN KEY (Child) REFERENCES Persons(ID)," +
					"FOREIGN KEY (Father) REFERENCES Persons(ID)," +
					"FOREIGN KEY (Mother) REFERENCES Persons(ID))" ;
			myStmt.executeUpdate(family);
			System.out.println("Created table Family in given database...");

			//Inserting the values of child and parent in the table
			String finsert = "INSERT INTO Family VALUES(1, 5, 4)";
			myStmt.executeUpdate(finsert);
			finsert = "INSERT INTO Family VALUES(2, 5, 4)";
			myStmt.executeUpdate(finsert);
			finsert = "INSERT INTO Family VALUES(3, 5, 4)";
			myStmt.executeUpdate(finsert);
			finsert = "INSERT INTO Family VALUES(6, 8, 2)";
			myStmt.executeUpdate(finsert);
			finsert = "INSERT INTO Family VALUES(7, 8, 2)";
			myStmt.executeUpdate(finsert);
			finsert = "INSERT INTO Family VALUES(5, 10, 9)";
			myStmt.executeUpdate(finsert);
			finsert = "INSERT INTO Family VALUES(4, 12, 11)";
			myStmt.executeUpdate(finsert);
			finsert = "INSERT INTO Family VALUES(14, 1, 13)";
			myStmt.executeUpdate(finsert);
			System.out.println("Inserted records into the table family...");

			//Here the results of the table are printed out.
			ResultSet myRs = myStmt.executeQuery("select * from Family");
			while (myRs.next()) {
				int child  = myRs.getInt("Child");
				int father = myRs.getInt("Father");
				int mother  = myRs.getInt("Mother");

				System.out.println("Child:" + child +
								", Father:" + father +
								", Mother:" + mother);
```

```
        }

        /*
         * A stored procedure is created to give the children from the parents inputed
         * In the procedure we are using the tables Persons and Family to find the
           children
        */
        String children="CREATE PROCEDURE ChildrenOf(IN Mother INT, IN Father INT)\n"+
                    "BEGIN\r\n" +
                    "    SELECT Family.Child, Persons.Name, Persons.Gender\r\n" +
                    "    FROM Family\r\n" +
                    "    JOIN Persons\r\n" +
                    "    ON Persons.ID = Family.Child\r\n" +
                    "    WHERE Family.Mother = (SELECT ID FROM Persons WHERE ID =
                                                Mother) \r\n" +
                    "          AND Family.Father = (SELECT ID FROM Persons WHERE ID =
                                                   Father);\r\n" +
                    "END";
          myStmt.execute(children);
       System.out.println("Procedure ChildrenOf Created......");

       String getChildren = "{CALL ChildrenOf(?, ?)}";
       CallableStatement stmt = myConn.prepareCall(getChildren);

       /* Once the procedure is created, we are going to call the procedure ChildrenOf
       In call we use the values 4 and 5 which are Raquel and Giselo */
       stmt.setInt(1, 4);
       stmt.setInt(2, 5);
       rs = stmt.executeQuery();
       System.out.println("\r\n" +
                                    "The children of Raquel and Giselo");
while (rs.next()) {
    System.out.println(String.format("ID:" + rs.getInt("Child") +
                                     ", Name:" + rs.getString("Name") +
                                     ", Gender:" + rs.getString("Gender")));
}
System.out.println("Stored procedure called successfully!");

/* Here we are calling the procedure again using the values 2 and 8
Which will give us the children of Nancy and Chris */
stmt.setInt(1, 2);
       stmt.setInt(2, 8);
       rs = stmt.executeQuery();
       System.out.println("\r\n" +
                          "The children of Nancy and Chris");
while (rs.next()) {
    System.out.println(String.format("ID:" + rs.getInt("Child") +
                                     ", Name:" + rs.getString("Name") +
                                     ", Gender:" + rs.getString("Gender")));
}
System.out.println("Stored procedure called successfully!");

// This will gives us the children Lisa and Jason and their ID's are 13 and 1
stmt.setInt(1, 13);
       stmt.setInt(2, 1);
       rs = stmt.executeQuery();
       System.out.println("\r\n" +
                          "The children of Lisa and Jason");
```

```
            while (rs.next()) {
               System.out.println(String.format("ID:" + rs.getInt("Child") +
                                                ", Name:" + rs.getString("Name") +
                                                ", Gender:" + rs.getString("Gender")));
            }
            System.out.println("Stored procedure called successfully!");
             }
             catch (Exception exc) {
                    exc.printStackTrace();
             }

      }

}
```

GrandParents.java

```
package relations;
import java.sql.*;

public class GrandParent extends Persons {

	public static void main(String[] args) {

		String url = "jdbc:mysql://localhost:3306/people";
		String username = "root";
		String password = "Something11";

		try {
			//Connecting to the database to input the data
			System.out.println("Connecting to a selected database...");
			Connection myConn = DriverManager.getConnection(url, username, password);
			System.out.println("Connected database successfully...");
			Statement myStmt = myConn.createStatement();
			ResultSet rs;

			//Creating the table GP to see the grandparents of each person listed
			String gp = "CREATE TABLE GP (\r\n" +
					"    GrandChild INT, \r\n" +
					"    GrandMother INT, \r\n" +
					"    GrandFather INT, \r\n" +
					"    FOREIGN KEY (Grandchild) REFERENCES Persons(ID),\r\n" +
					"    FOREIGN KEY (GrandMother) REFERENCES Persons(ID),\r\n" +
					"    FOREIGN KEY (GrandFather) REFERENCES Persons(ID)\r\n" +
					")";
			myStmt.executeUpdate(gp);
			System.out.println("Created table GP in given database...");

			//Inputting each of the values for the GP table
			String gpinsert = "INSERT INTO GP VALUES(6, 4, 5)";
			myStmt.executeUpdate(gpinsert);
			gpinsert = "INSERT INTO GP VALUES(7, 4, 5)";
			myStmt.executeUpdate(gpinsert);
			gpinsert = "INSERT INTO GP VALUES(1, 9, 10)";
			myStmt.executeUpdate(gpinsert);
			gpinsert = "INSERT INTO GP VALUES(1, 11, 12)";
			myStmt.executeUpdate(gpinsert);
			gpinsert = "INSERT INTO GP VALUES(2, 9, 10)";
			myStmt.executeUpdate(gpinsert);
			gpinsert = "INSERT INTO GP VALUES(2, 11, 12)";
			myStmt.executeUpdate(gpinsert);
			gpinsert = "INSERT INTO GP VALUES(3, 9, 10)";
			myStmt.executeUpdate(gpinsert);
			gpinsert = "INSERT INTO GP VALUES(3, 11, 12)";
			myStmt.executeUpdate(gpinsert);
			System.out.println("Inserted records into the table GP...");

			//Here the results of the table are printed out.
			ResultSet myRs = myStmt.executeQuery("select * from GP");
			while (myRs.next()) {
				int gc  = myRs.getInt("GrandChild");
				int gm = myRs.getInt("GrandMother");
				int gf  = myRs.getInt("GrandFather");
```

```
            System.out.println("GrandChild:" + gc +
                               ", GrandMother:" + gm +
                               ", GrandFather:" + gf);
        }

        /*
         * The procedure is created that would give us the grandparents of the person
           inputed.
         * In this procedure we are using the tables GP and Persons
        */
        String grandparents = "CREATE PROCEDURE GrandOf(IN gc INT)\r\n" +
                    "BEGIN\r\n" +
                    "      SELECT Persons.ID, Persons.Name FROM GP\r\n" +
                    "    JOIN Persons\r\n" +
                    "    ON Persons.ID = GP.GrandMother OR
                         Persons.ID = GP.GrandFather\r\n" +
                    "WHERE GP.GrandChild = (SELECT ID FROM Persons WHERE ID=gc);\n" +
                    "END";
       myStmt.execute(grandparents);
     System.out.println("Procedure GrandOf Created......");

     // Establishing the way to call the procedure  GrandOf
     String getGrand = "{CALL GrandOf(?)}";
     CallableStatement stmt = myConn.prepareCall(getGrand);

     //Calling the procedure with the input of 1 which is Jason
     stmt.setInt(1, 1);
     rs = stmt.executeQuery();
     System.out.println("\r\n" +
                                        "The grandparents of Jason");
while (rs.next()) {
    System.out.println(String.format("ID: " + rs.getInt("ID") +
                                     ", GrandParent:" + rs.getString("Name")));
}
System.out.println("Stored procedure called successfully!");

// Calling the procedure again but with the input of 2 which is Nancy
stmt.setInt(1, 2);
 rs = stmt.executeQuery();
 System.out.println("\r\n" +
                    "The grandparents of Nancy");
while (rs.next()) {
    System.out.println(String.format("ID: " + rs.getInt("ID") +
                                     ", GrandParent:" + rs.getString("Name")));
}
System.out.println("Stored procedure called successfully!");

// Calling the procedure one more time for the input of 6, it is Evelyn
stmt.setInt(1, 6);
     rs = stmt.executeQuery();
     System.out.println("\r\n" +
                        "The grandparents of Evelyn");
while (rs.next()) {
    System.out.println(String.format("ID: " + rs.getInt("ID") +
                                     ", GrandParent:" + rs.getString("Name")));
}
System.out.println("Stored procedure called successfully!");

 }
 catch (Exception exc) {
        exc.printStackTrace();
```

```
		}

	}

}
```

SisterInLaw.java

```
package relations;
import java.sql.*;

public class SsiterInLaw {

	public static void main(String[] args) {
		String url = "jdbc:mysql://localhost:3306/people";
		String username = "root";
		String password = "Something11";

		try {
			// Connecting to the database we are using in MySQL workbench
			System.out.println("Connecting to a selected database...");
			Connection myConn = DriverManager.getConnection(url, username, password);
			System.out.println("Connected database successfully...");
			Statement myStmt = myConn.createStatement();
			ResultSet rs;

			// The table brother is created showing the brother of a person
			String brother = "CREATE TABLE brother (\r\n" +
					"       Brother INT,\r\n" +
					"    Person INT,\r\n" +
					"    FOREIGN KEY (Brother) REFERENCES Persons(ID),\r\n" +
					"    FOREIGN KEY (Person) REFERENCES Persons(ID)\r\n" +
					")";
			myStmt.executeUpdate(brother);
			System.out.println("Created table brother in given database...");

			//Inserting the values for brothers to a person
			String binsert = "INSERT INTO brother VALUES(1, 2)";
			myStmt.executeUpdate(binsert);
			binsert = "INSERT INTO brother VALUES(1, 3)";
			myStmt.executeUpdate(binsert);
			binsert = "INSERT INTO brother VALUES(3, 1)";
			myStmt.executeUpdate(binsert);
			binsert = "INSERT INTO brother VALUES(3, 2)";
			myStmt.executeUpdate(binsert);
			binsert = "INSERT INTO brother VALUES(7, 6)";
			myStmt.executeUpdate(binsert);
			System.out.println("Inserted records into the table brother...");

			//Then the sister table has been created, showing the sister of a person
			String sister = "CREATE TABLE sister (\r\n" +
					"       Sister INT,\r\n" +
					"    Person INT,\r\n" +
					"    FOREIGN KEY (Sister) REFERENCES Persons(ID),\r\n" +
					"    FOREIGN KEY (Person) REFERENCES Persons(ID)\r\n" +
					");";
			myStmt.executeUpdate(sister);
			System.out.println("Created table sister in given database...");
```

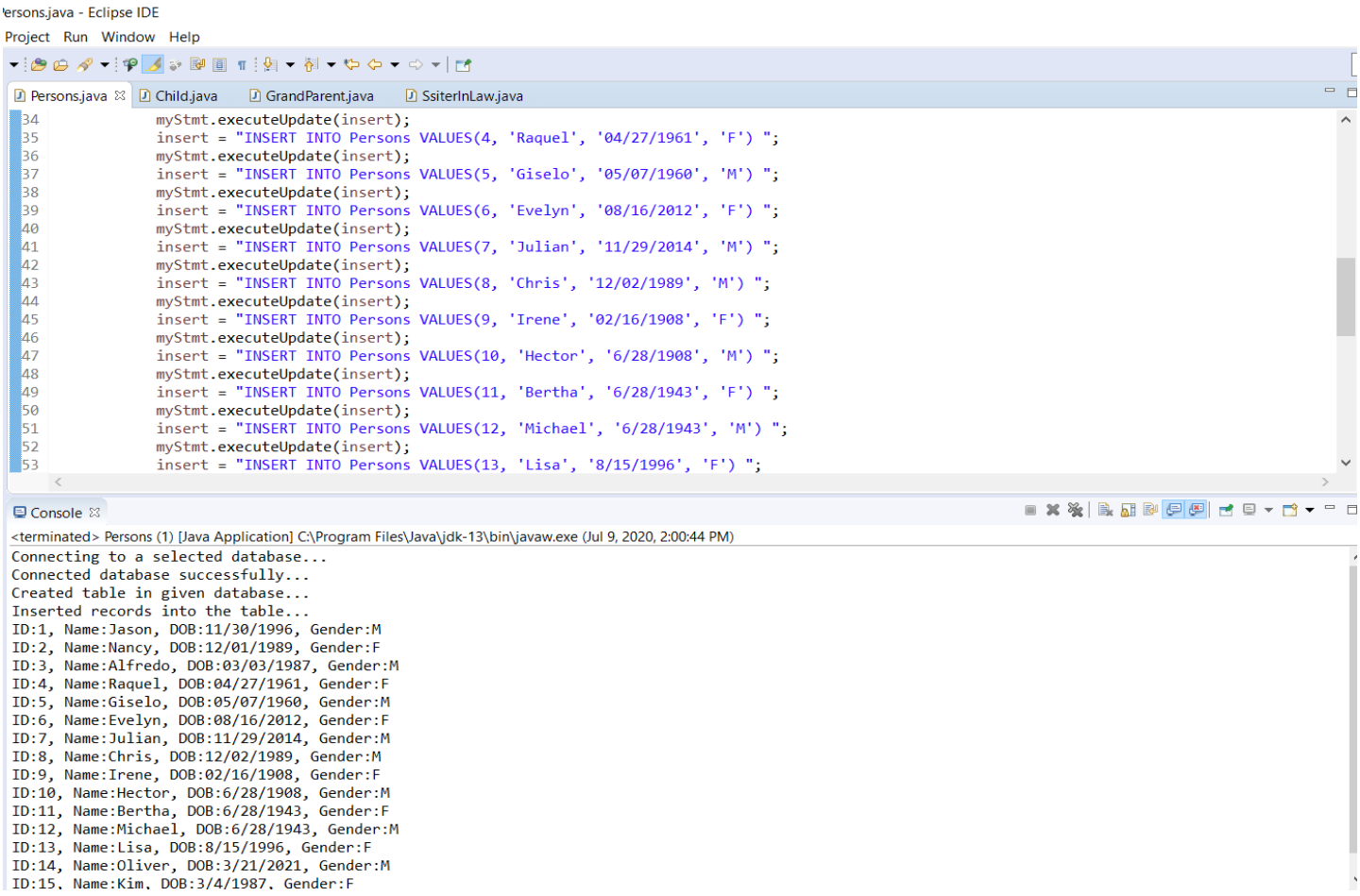
```
// The values are being inserted into the table sister
String sinsert = "INSERT INTO sister VALUES(2, 1)";
myStmt.executeUpdate(sinsert);
sinsert = "INSERT INTO sister VALUES(2, 3)";
myStmt.executeUpdate(sinsert);
sinsert = "INSERT INTO sister VALUES(6, 7)";
myStmt.executeUpdate(sinsert);
System.out.println("Inserted records into the table sister...");

// Creating the table for bro_sis, which will provide us only the brother and
   sister
String bro_sis = "CREATE TABLE bro_sis (\r\n" +
             "       Brother INT,\r\n" +
             "    Sister INT,\r\n" +
             "    FOREIGN KEY (Brother) REFERENCES Persons(ID),\r\n" +
             "    FOREIGN KEY (Sister) REFERENCES Persons(ID)\r\n" +
             ")";
myStmt.executeUpdate(bro_sis);
System.out.println("Created table bro_sis in given database...");

//Inserting the values for the table providing the brother/sister relationship
String bsinsert = "INSERT INTO bro_sis VALUES(1, 2)";
myStmt.executeUpdate(bsinsert);
bsinsert = "INSERT INTO bro_sis VALUES(3, 2)";
myStmt.executeUpdate(bsinsert);
bsinsert = "INSERT INTO bro_sis VALUES(7, 6)";
myStmt.executeUpdate(bsinsert);

/*
 * Now the table of husband and wife is created
 * This will show for every person there will be a husband or wife
 */
String hus_wif = "CREATE TABLE hus_wif (\r\n" +
             "       Husband INT,\r\n" +
             "    Wife INT,\r\n" +
             "    FOREIGN KEY (Husband) REFERENCES Persons(ID),\r\n" +
             "    FOREIGN KEY (Wife) REFERENCES Persons(ID)\r\n" +
             ")";
myStmt.executeUpdate(hus_wif);
System.out.println("Created table hus_wif in given database...");

// The values of the table hus_wif are being inserted
String hwinsert = "INSERT INTO hus_wif VALUES(1, 13)";
myStmt.executeUpdate(hwinsert);
hwinsert = "INSERT INTO hus_wif VALUES(8, 2)";
myStmt.executeUpdate(hwinsert);
hwinsert = "INSERT INTO hus_wif VALUES(5, 4)";
myStmt.executeUpdate(hwinsert);
hwinsert = "INSERT INTO hus_wif VALUES(10, 9)";
myStmt.executeUpdate(hwinsert);
hwinsert = "INSERT INTO hus_wif VALUES(12, 11)";
myStmt.executeUpdate(hwinsert);
hwinsert = "INSERT INTO hus_wif VALUES(3, 15)";
myStmt.executeUpdate(hwinsert);

/*
 * Creating the table sislaw
 * Which will contain the sister in law for a given person
 */
String sislaw = "CREATE TABLE sislaw (\r\n" +
             "       Sis_Inlaw INT,\r\n" +
```

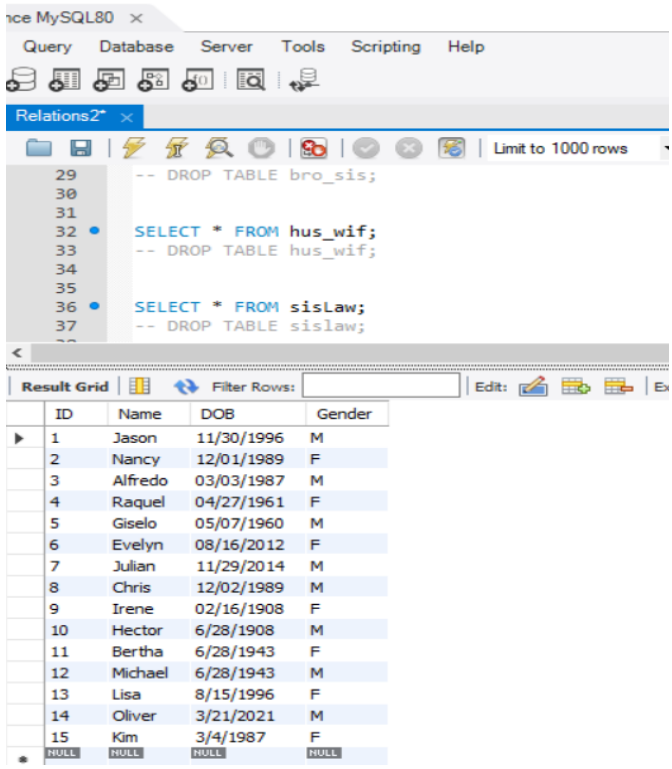
```
                "    Person INT,\r\n" +
                "    FOREIGN KEY (Sis_Inlaw) REFERENCES Persons(ID),\r\n" +
                "    FOREIGN KEY (Person) REFERENCES Persons(ID)\r\n" +
                ")";
        myStmt.executeUpdate(sislaw);
        System.out.println("Created table sislaw in given database...");

        /*
         * We are inserting the values to the table but not like the previous tables
         * Here we are using the tables hus_wif and brother to find
         * relationships between the wife and siblings of their husband.
         */
        String slinsert = "INSERT INTO sislaw(Sis_Inlaw, Person)\r\n" +
                "SELECT hus_wif.Wife, brother.Person FROM hus_wif\r\n" +
                "JOIN brother ON hus_wif.Husband = brother.Brother\r\n" +
                "WHERE hus_wif.Wife = 13 OR hus_wif.Wife = 15";
        myStmt.executeUpdate(slinsert);

        /*
         * In this we are inserting the values using the tables hus_wif and sister
         * This relationship will find a sister and a wife, check if they share a
           person
         * which is in the columns hus_wif.husband/sister.person
         */
        slinsert = "INSERT INTO sislaw(Sis_Inlaw, Person)\r\n" +
                "SELECT sister.Sister, hus_wif.Wife FROM hus_wif\r\n" +
                "JOIN sister ON hus_wif.Husband = sister.Person\r\n" +
                "WHERE hus_wif.Husband = 1 OR hus_wif.Husband = 3;";
        myStmt.executeUpdate(slinsert);

        //Since we got our information from the previous statements. We will see the
          data
        ResultSet myRs = myStmt.executeQuery("select * from sislaw");
        while (myRs.next()) {
              int inlaw  = myRs.getInt("Sis_Inlaw");
              int person = myRs.getInt("Person");

              System.out.println("Sis_Inlaw:" + inlaw +
                                        ", Person:" + person);
        }

        /*
         * Now creating a procedure to receive the sister in law of a given person
         * In the procedure we are using the tables Persons and sislaw to get their
         * ID, names and their gender.
         */
        String inlawOf = "CREATE PROCEDURE inlawOf(IN Person INT)\r\n" +
                "BEGIN\r\n" +
                "SELECT Persons.ID, Persons.Name, Persons.Gender FROM sislaw\n" +
                "    JOIN Persons\r\n" +
                "    ON Persons.ID = sislaw.Sis_Inlaw\r\n" +
                "    WHERE sislaw.Person = (SELECT Persons.ID FROM Persons
                                            WHERE ID = Person);\r\n" +
                "END";
        myStmt.execute(inlawOf);
    System.out.println("Procedure inlawOf Created......");

    //Here we are establishing how to call the procedure that has been created
    String getInlaw = "{CALL inlawOf(?)}";
    CallableStatement stmt = myConn.prepareCall(getInlaw);
```

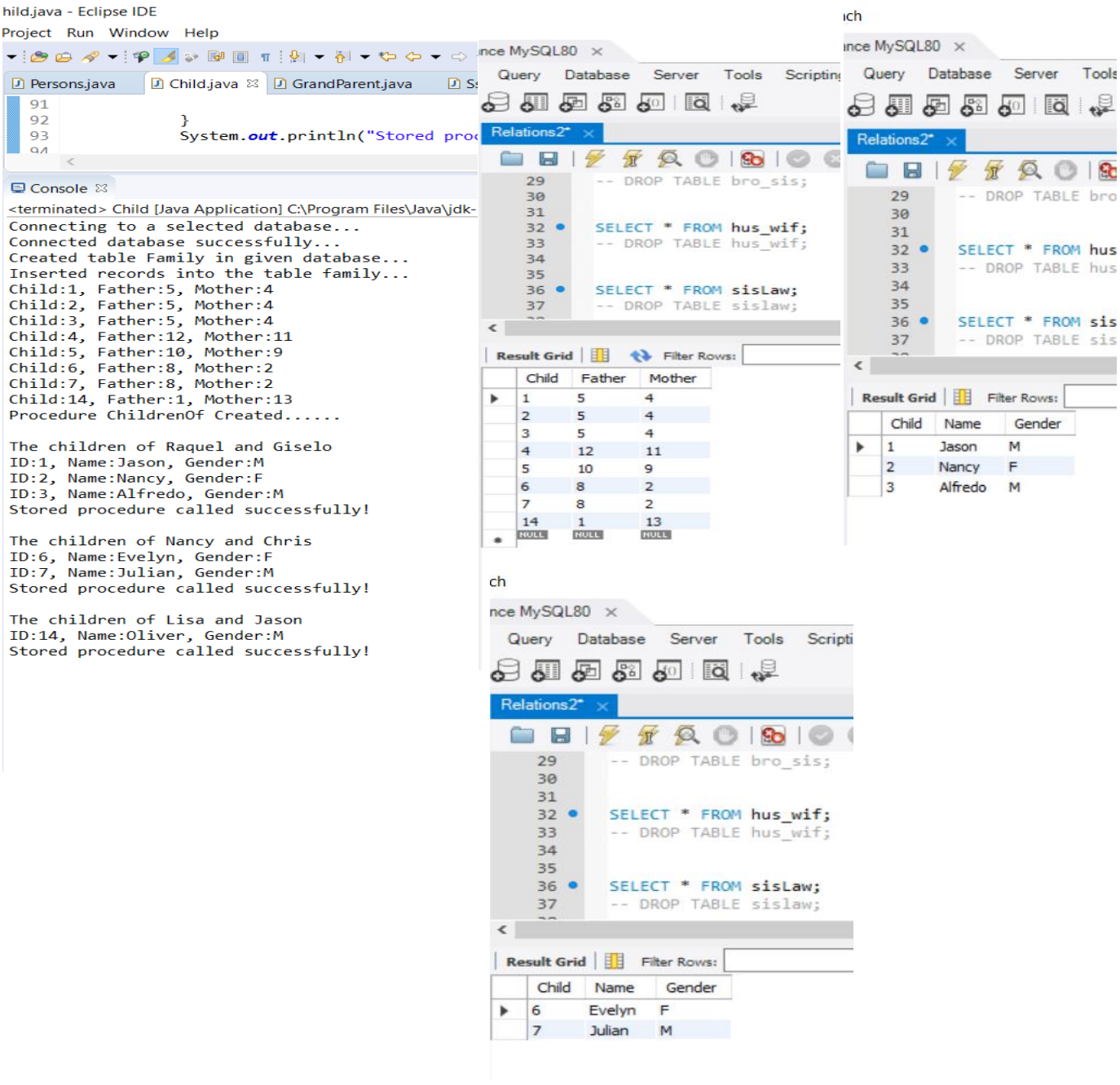
```
            //Now we are giving a value to the procedure, it is 2 which is Nancy
            stmt.setInt(1, 2);
            rs = stmt.executeQuery();
            System.out.println("\r\n" +
                               "The Sisters in law of Nancy");
          while (rs.next()) {
              System.out.println(String.format("ID:" + rs.getInt("ID") +
                                               ", Name:" + rs.getString("Name") +
                                               ", Gender:" + rs.getString("Gender")));
          }
          System.out.println("Stored procedure called successfully!");

        //Now we are giving a value to the procedure, it is 13 which is Lisa
          stmt.setInt(1, 13);
               rs = stmt.executeQuery();
               System.out.println("\r\n" +
                               "The Sisters in law of Lisa");
          while (rs.next()) {
              System.out.println(String.format("ID:" + rs.getInt("ID") +
                                               ", Name:" + rs.getString("Name") +
                                               ", Gender:" + rs.getString("Gender")));
          }
          System.out.println("Stored procedure called successfully!");

        //Now we are giving a value to the procedure, it is 1 which is Jason
          stmt.setInt(1, 1);
               rs = stmt.executeQuery();
               System.out.println("\r\n" +
                               "The Sisters in law of Jason");
          while (rs.next()) {
              System.out.println(String.format("ID:" + rs.getInt("ID") +
                                               ", Name:" + rs.getString("Name") +
                                               ", Gender:" + rs.getString("Gender")));
          }
          System.out.println("Stored procedure called successfully!");
           }
           catch (Exception exc) {
                  exc.printStackTrace();
           }
    }

}
```

Output:

I am showing the outputs from the java application and from MySQL to show it's the same results.






hild.java - Eclipse IDE
Project Run Window Help
Persons.java
Child.java
GrandParent.java
91
92 }
93 System.out.println("Stored proc
Console
<terminated> Child [Java Application] C:\Program Files\Java\jdk-
Connecting to a selected database...
Connected database successfully...
Created table Family in given database...
Inserted records into the table family...
Child:1, Father:5, Mother:4
Child:2, Father:5, Mother:4
Child:3, Father:5, Mother:4
Child:4, Father:12, Mother:11
Child:5, Father:10, Mother:9
Child:6, Father:8, Mother:2
Child:7, Father:8, Mother:2
Child:14, Father:1, Mother:13
Procedure ChildrenOf Created.....
The children of Raquel and Giselo
ID:1, Name:Jason, Gender:M
ID:2, Name:Nancy, Gender:F
ID:3, Name:Alfredo, Gender:M
Stored procedure called successfully!
The children of Nancy and Chris
ID:6, Name:Evelyn, Gender:F
ID:7, Name:Julian, Gender:M
Stored procedure called successfully!
The children of Lisa and Jason
ID:14, Name:Oliver, Gender:M
Stored procedure called successfully!
nce MySQL80
Query Database Server Tools Scripting
Relations2*
29 -- DROP TABLE bro_sis;
30
31
32 SELECT * FROM hus_wif;
33 -- DROP TABLE hus_wif;
34
35
36 SELECT * FROM sisLaw;
37 -- DROP TABLE sislaw;
Result Grid Filter Rows:
Child Father Mother
1 5 4
2 5 4
3 5 4
4 12 11
5 10 9
6 8 2
7 8 2
14 1 13
NULL NULL NULL
ch
nce MySQL80
Query Database Server Tools Scripti
Relations2*
29 -- DROP TABLE bro_sis;
30
31
32 SELECT * FROM hus_wif;
33 -- DROP TABLE hus_wif;
34
35
36 SELECT * FROM sisLaw;
37 -- DROP TABLE sislaw;
Result Grid Filter Rows:
Child Name Gender
6 Evelyn F
7 Julian M
ch
nce MySQL80
Query Database Server Tools
Relations2*
29 -- DROP TABLE bro
30
31
32 SELECT * FROM hus
33 -- DROP TABLE hus
34
35
36 SELECT * FROM sis
37 -- DROP TABLE sis
Result Grid Filter Rows:
Child Name Gender
1 Jason M
2 Nancy F
3 Alfredo M

GrandParent.java - Eclipse IDE

Project Run Window Help

Persons.java | Child.java | GrandParent.java | SsiterIn

```
91
92              // Calling the procedure again but
93              stmt.setInt(1, 2);
94              rs = stmt.executeQuery();
```

Console

<terminated> GrandParent [Java Application] C:\Program Files\Java\jdk

```
Connecting to a selected database...
Connected database successfully...
Created table GP in given database...
Inserted records into the table GP...
GrandChild:6, GrandMother:4, GrandFather:5
GrandChild:7, GrandMother:4, GrandFather:5
GrandChild:1, GrandMother:9, GrandFather:10
GrandChild:1, GrandMother:11, GrandFather:12
GrandChild:2, GrandMother:9, GrandFather:10
GrandChild:2, GrandMother:11, GrandFather:12
GrandChild:3, GrandMother:9, GrandFather:10
GrandChild:3, GrandMother:11, GrandFather:12
Procedure GrandOf Created......

The grandparents of Jason
ID: 9, GrandParent:Irene
ID: 10, GrandParent:Hector
ID: 11, GrandParent:Bertha
ID: 12, GrandParent:Michael
Stored procedure called successfully!

The grandparents of Nancy
ID: 9, GrandParent:Irene
ID: 10, GrandParent:Hector
ID: 11, GrandParent:Bertha
ID: 12, GrandParent:Michael
Stored procedure called successfully!

The grandparents of Evelyn
ID: 4, GrandParent:Raquel
ID: 5, GrandParent:Giselo
Stored procedure called successfully!
```

ch

nce MySQL80

Query Database Server Tools Scripting

Relations2*

```
29  -- DROP TABLE bro_sis;
30
31
32  SELECT * FROM hus_wif;
33  -- DROP TABLE hus_wif;
34
35
36  SELECT * FROM sisLaw;
37  -- DROP TABLE sislaw;
```

Result Grid | Filter Rows:

GrandChild	GrandMother	GrandFather
6	4	5
7	4	5
1	9	10
1	11	12
2	9	10
2	11	12
3	9	10
3	11	12

ch

nce MySQL80

Query Database Server Tools

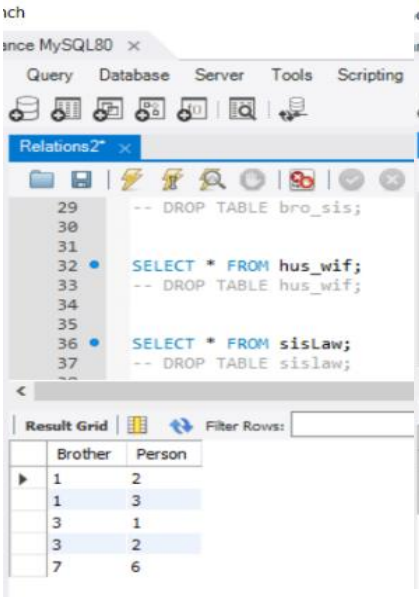
Relations2*

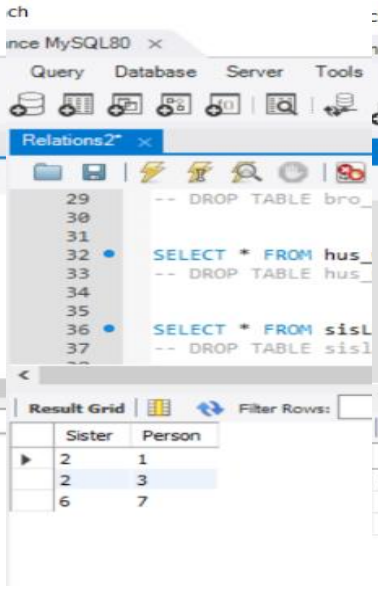
```
29  -- DROP TABLE bro_s
30
31
32  SELECT * FROM hus_w
33  -- DROP TABLE hus_w
34
35
36  SELECT * FROM sisLa
37  -- DROP TABLE sisla
```

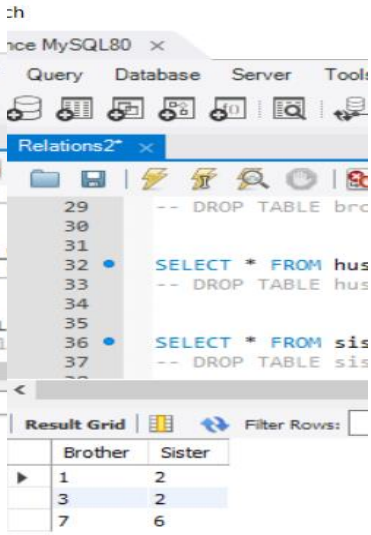
Result Grid | Filter Rows:

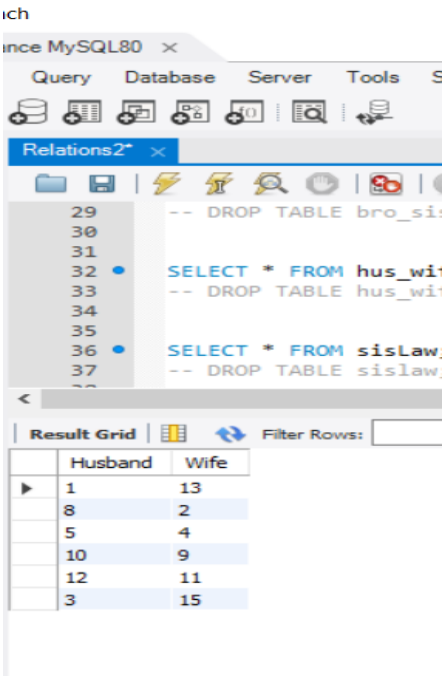
ID	Name
9	Irene
10	Hector
11	Bertha
12	Michael

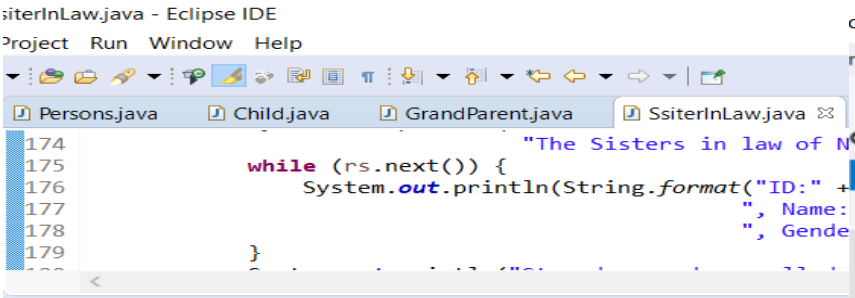
Output of the other tables that were not shown in the java application.

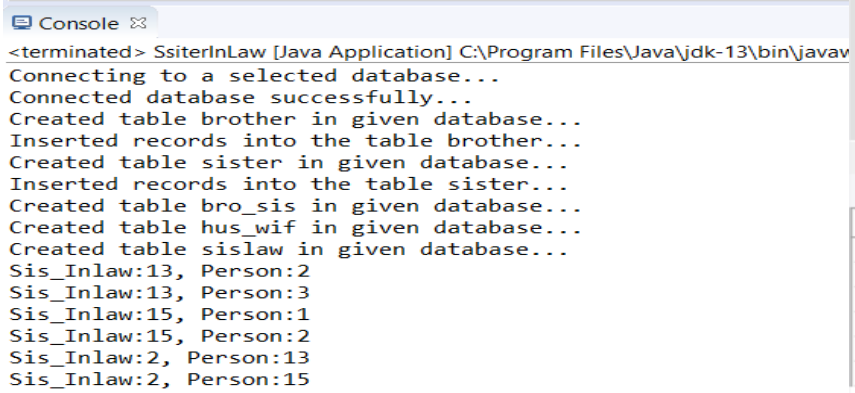


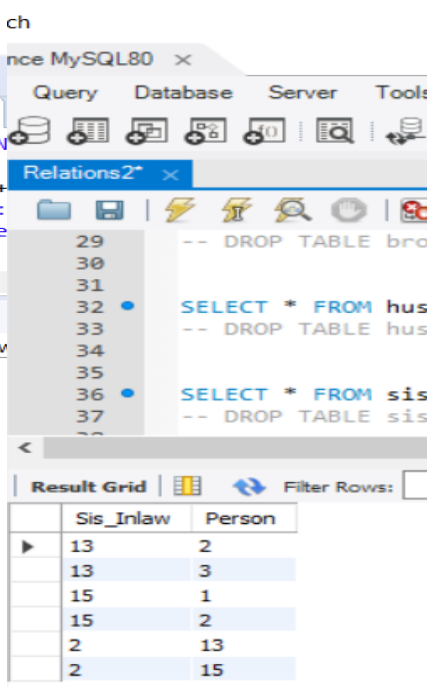


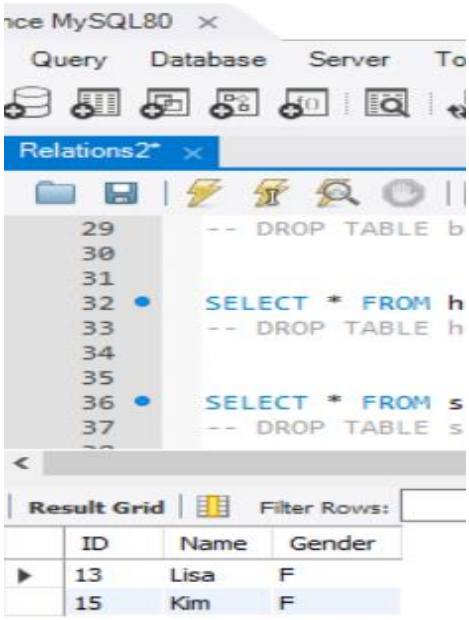





siterInLaw.java - Eclipse IDE
Project Run Window Help
Persons.java
Child.java
GrandParent.java
SsiterInLaw.java
174 "The Sisters in law of N
175 while (rs.next()) {
176 System.out.println(String.format("ID:" +
177 ", Name:
178 ", Gende
179 }


Console
<terminated> SsiterInLaw [Java Application] C:\Program Files\Java\jdk-13\bin\javaw
Connecting to a selected database...
Connected database successfully...
Created table brother in given database...
Inserted records into the table brother...
Created table sister in given database...
Inserted records into the table sister...
Created table bro_sis in given database...
Created table hus_wif in given database...
Created table sislaw in given database...
Sis_Inlaw:13, Person:2
Sis_Inlaw:13, Person:3
Sis_Inlaw:15, Person:1
Sis_Inlaw:15, Person:2
Sis_Inlaw:2, Person:13
Sis_Inlaw:2, Person:15
Procedure inlawOf Created.....
The Sisters in law of Nancy
ID:13, Name:Lisa, Gender:F
ID:15, Name:Kim, Gender:F
Stored procedure called successfully!
The Sisters in law of Lisa
ID:2, Name:Nancy, Gender:F
Stored procedure called successfully!
The Sisters in law of Jason
ID:15, Name:Kim, Gender:F
Stored procedure called successfully!


ch
nce MySQL80
Query Database Server Tools
Relations2*
29 -- DROP TABLE bro
30
31
32 SELECT * FROM hus
33 -- DROP TABLE hus
34
35
36 SELECT * FROM sis
37 -- DROP TABLE sis
Result Grid Filter Rows:
Sis_Inlaw Person
13 2
13 3
15 1
15 2
2 13
2 15


ch
nce MySQL80
Query Database Server To
Relations2*
29 -- DROP TABLE b
30
31
32 SELECT * FROM h
33 -- DROP TABLE h
34
35
36 SELECT * FROM s
37 -- DROP TABLE s
Result Grid Filter Rows:
ID Name Gender
13 Lisa F
15 Kim F