import java.sql.\*;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.util.Scanner;  
  
import static java.lang.System.*exit*;  
  
  
public class PostgreSQLJDBC {  
 public static Connection *c* = null;  
 public static Statement *stmt*=null;  
  
 public static void soyagaci(int id){  
 try {  
 *stmt*=*c*.createStatement();  
 ResultSet rs = *stmt*.executeQuery("select\*from person where id=" + id);  
  
 while (rs.next()) {  
 int pid = rs.getInt("id");  
 String name = rs.getString("name");  
  
 System.*out*.println("id: " + pid + " adi: " + name);  
  
 if (rs.getInt("fatherid") != 0)  
 *soyagaci*(rs.getInt("fatherid"));  
 if (rs.getInt("motherid") != 0)  
 *soyagaci*(rs.getInt("motherid"));  
  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
  
 public static void soyundangelen(int id){  
 try{  
 *stmt*=*c*.createStatement();  
 ResultSet rs = *stmt*.executeQuery("select \* from person where motherid::int="+id+" or fatherid::int="+ id);  
  
 while (rs.next()) {  
 int pid = rs.getInt("id");  
 String name = rs.getString("name");  
  
 System.*out*.println("id: " + pid + " adi: " + name);  
  
 if(rs.getInt("fatherid") != 0 || rs.getInt("motherid") != 0)  
 *soyundangelen*(rs.getInt("id"));  
  
 }  
  
 }catch(SQLException e){  
 e.printStackTrace();  
 }  
 }  
  
  
 public static void menu(){  
  
 while (true) {  
 System.*out*.println("Menu");  
 System.*out*.println("0: çıkış");  
 System.*out*.println("1: kardeş sorgula");  
 System.*out*.println("2: kuzen sorgula");  
 System.*out*.println("3: soyundan gelenleri sorgula");  
 System.*out*.println("4: soy ağacı sorgula");  
  
 System.*out*.println("Secim: ");  
  
 Scanner in = new Scanner(System.*in*);  
 int secim = in.nextInt();  
 int id;  
  
 switch (secim) {  
 case 0:  
 *exit*(0);  
 break;  
 case 1:  
 System.*out*.println("Id girin: ");  
 id = in.nextInt();  
  
 try{  
 String sql = "(select name from person where fatherid in(" +  
 "select fatherid from person where id="+id+"))" +  
 "union" +  
 "(select name from person where motherid in(" +  
 "select fatherid from person where id="+id+"));";  
 ResultSet rs = *stmt*.executeQuery(sql);  
  
 while (rs.next()) {  
 System.*out*.println(rs.getString(1));  
  
 }  
 }catch (SQLException e){  
 e.printStackTrace();  
 }  
  
 break;  
 case 2:  
 System.*out*.println("Id girin: ");  
 id = in.nextInt();  
  
 try{  
 String sql = "select\*from kuzen("+ id +");";  
 ResultSet rs = *stmt*.executeQuery(sql);  
  
 while (rs.next()) {  
 System.*out*.println(rs.getString(1));  
  
 }  
 }catch (SQLException e){  
 e.printStackTrace();  
 }  
 break;  
 case 3:  
 System.*out*.println("Id girin: ");  
 id = in.nextInt();  
 *soyundangelen*(id);  
  
 break;  
 case 4:  
 System.*out*.println("Id girin: ");  
 id = in.nextInt();  
 *soyagaci*(id);  
 break;  
 }  
 }  
 }  
  
 public static void main(String args[]) {  
  
 try {  
 Class.*forName*("org.postgresql.Driver");  
 *c* = DriverManager  
 .*getConnection*("jdbc:postgresql://localhost:5432/soy",  
 "postgres", "postgres");  
 *stmt* = *c*.createStatement();  
  
 *menu*();  
 }catch (Exception e) {  
 e.printStackTrace();  
 System.*err*.println(e.getClass().getName()+": "+e.getMessage());  
 *exit*(0);  
 }  
 System.*out*.println("Opened database successfully");  
  
  
}}

CREATE OR REPLACE FUNCTION "public"."kuzen"(int4)

RETURNS TABLE("name" varchar) AS $BODY$

declare r record;

d integer; a integer; bb integer; ba integer;

begin

for r in select\*from person

loop

select count(\*) into d from( select dedeid from ata($1) intersect select dedeid from ata(r.id)) as bir;

select count(\*) into a from( select anneanneid from ata($1) intersect select anneanneid from ata(r.id)) as iki;

select count(\*) into bb from( select buyukbabaid from ata($1) intersect select buyukbabaid from ata(r.id)) as uc;

select count(\*) into ba from( select buyukanneid from ata($1) intersect select buyukanneid from ata(r.id)) as dort;

if d != 0 or a != 0 or bb != 0 or ba != 0 then

return query(select distinct r.name from person except (

select p1.name from person p1 where motherid in(select p.motherid from person p where p.id=$1)

union

select p2.name from person p2 where fatherid in(select p3.fatherid from person p3 where p3.id=$1)

)

);

end if;

end loop;

end;

$BODY$

LANGUAGE plpgsql

CREATE OR REPLACE FUNCTION "public"."ata"(int4)

RETURNS "public"."my\_type" AS $BODY$

declare

fid int;

mid int; result\_record my\_type;

begin

select fatherid,motherid into fid,mid from person where id=$1;

select fatherid,motherid into result\_record.buyukbabaid, result\_record.buyukanneid from person where id=fid;

select fatherid,motherid into result\_record.dedeid,result\_record.anneanneid from person where id=mid;

return result\_record;

end;

$BODY$

LANGUAGE plpgsql VOLATILE

COST 100