Name:- Bhavesh Kewalramanis

Roll No.:- 25

Batch:- A1

Semester:-4th

Shift:- 1st

Section:- A

PRACTICAL-9

Solution:-

```
package com.mycompany.employeeproject;


import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collection;
import java.util.Collections;
import java.util.Comparator;
import java.util.Iterator;
import java.util.List;
import java.util.TreeSet;

class salarySort implements Comparator<Person> {
        public int compare(Person o1, Person o2) {
                return o1.getSalary().compareTo(o2.getSalary());
        }
}

class eidSort implements Comparator<Person> {

        public int compare(Person o1, Person o2) {
                return o1.getId().compareTo(o2.getId());
        }

}

class yoeSort implements Comparator<Person> {
        public int compare(Person o1, Person o2) {
                return o1.getYOE().compareTo(o2.getYOE());
        }
}

class multipleSort implements Comparator<Person> {
        private List<Comparator<Person>> listComparators;
```

```
    public multipleSort(Comparator<Person>...comparators) {
        this.listComparators = Arrays.asList(comparators);
    }

    public int compare(Person emp1, Person emp2) {
        for (Comparator<Person> comparator : listComparators) {
            int result = comparator.compare(emp1, emp2);
            if (result != 0) {
                return result;
            }
        }
        return 0;
    }
}


class Person {
        int aid;
        int eid;
          String name;
          int age;
          boolean isComorbid;
          int yoe;
```

```
      int salary;


       public Person() {


       }
       public Person(int eid,int fmaid,int yoe,int salary){
      this.eid=eid;
      this.aid=fmaid;
      this.yoe=yoe;
      this.salary=salary;
   }
      public Person(int eid,int aid,String name,int age,boolean
isComorbid,int yoe,int salary) {
               this.eid=eid;
            this.aid=aid;
            this.name=name;
            this.age=age;
            this.isComorbid=isComorbid;
            this.yoe=yoe;
            this.salary=salary;
       }
       public Person(int aid,String name,int age,boolean isComorbid) {
            this.aid=aid;
               this.name=name;
```

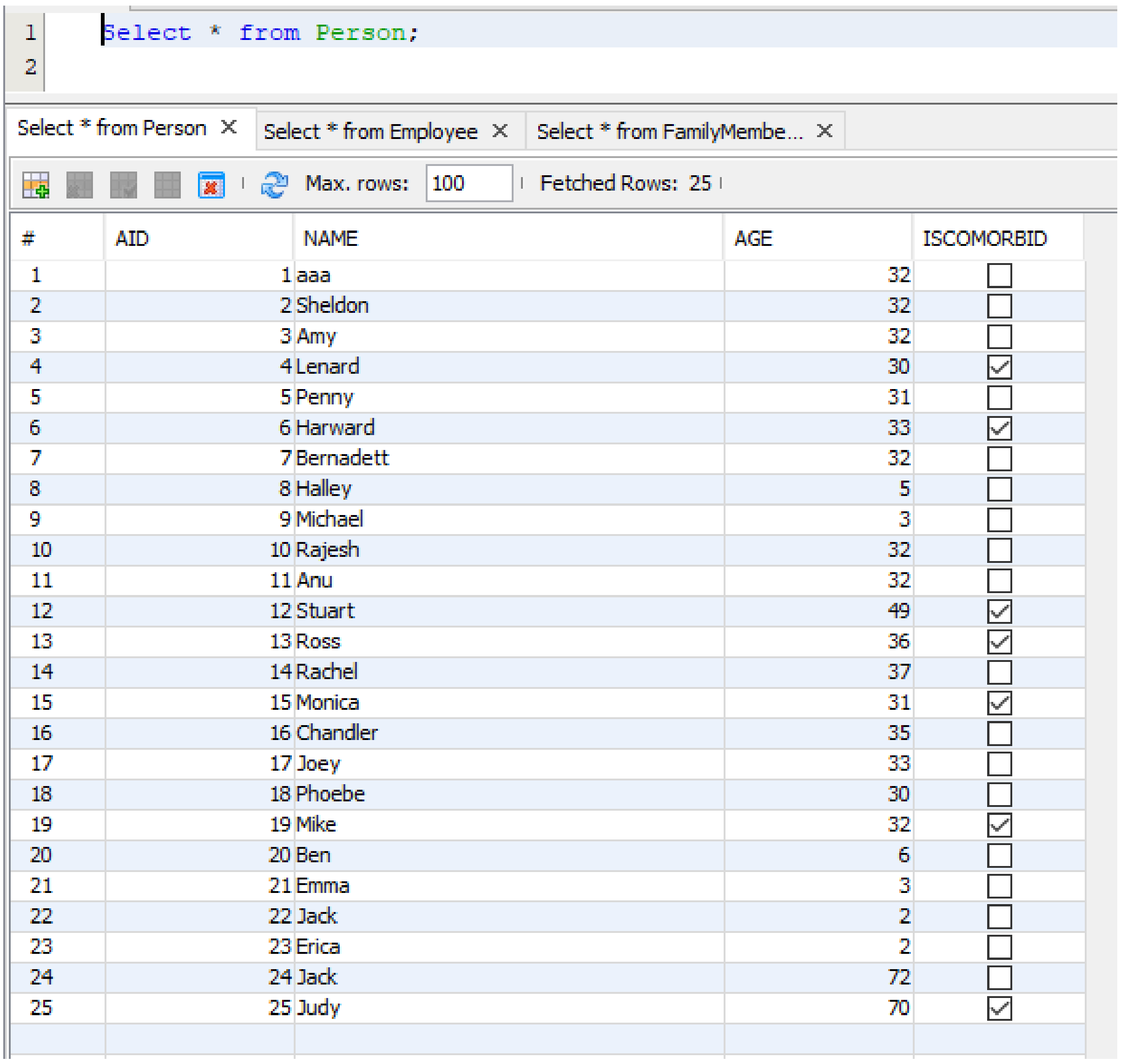
```
                this.age=age;

                this.isComorbid=isComorbid;

         }


       public Person(int eid, int aid, String name, int age, boolean
comorbid) {

              this.eid=eid;

              this.aid=aid;

                this.name=name;

                this.age=age;

                this.isComorbid=isComorbid;


   }

        public Integer getId() {

               return this.eid;

         }


         public Integer getYOE() {

                return this.yoe;

         }


         public Integer getSalary() {

                return this.salary;

         }
        @Override
         public String toString() {
                     return "Employee Id : "+this.eid+"\t"+"Family Member
Aadhar id : "+this.aid+"\t"+"Family Member Name :
"+this.name+"\t"+"Family Member Age : "+this.age+"\t\t"+"Is Family
Member Comorbid : "+this.isComorbid+"\n";
          }


        public int vaccine(int age,boolean isComorbid){
            if(age>=60){
                return 1;
            }else if(age >=45 && age<60){
                if(isComorbid){
                    return 1;
                }else{
                    return 0;
                }
            }else{
                return 0;
            }
        }
}
class Employee{
        int aid;
         int eid;
         int salary;
         int yoe;
         List<Person> fms;
         Person[] fmss;

         public Employee() {

         }
         public Employee(int aid,int eid,int salary,int yoe) {
              this.aid=aid;
                 this.eid=eid;
                 this.salary=salary;
                 this.yoe=yoe;
         }
         public Employee(int aid,int eid,int salary,int
yoe,ArrayList<Person>fms) {
              this.aid=aid;
                 this.eid=eid;
                 this.salary=salary;
                 this.yoe=yoe;
                 this.fms=fms;
         }
        public Integer getId() {
              return this.eid;
        }

        public Integer getYOE() {
                return this.yoe;
        }

        public Integer getSalary() {
                return this.salary;
        }


        public ArrayList<Person> vaacine(Employee e) {
                        Person[] p=e.fms.toArray(new Person[0]);
                        ArrayList<Person> p1 =new ArrayList<>();
                        for(int i=1;i<=p.length;i++) {
                                if(p[i].age>=60) {
                                        p1.add(p[i]);
                                }
                                if((p[i].age<60 && p[i].age>45 &&
p[i].isComorbid==true)) {
                                        p1.add(p[i]);
```

```
                              }
                         }
                         return p1;
                    }
}




public class Department {
  String dment;
       TreeSet<Employee> ems;

       public Department() {

       }

       public Department(String dment,TreeSet<Employee>ems) {
              this.dment=dment;
              this.ems=ems;
       }

     public TreeSet<Employee> getemployees(){
     return ems;
  }
```

```
   public static void main(String[] args) {
      Connection conn = null;
      Statement stmt = null;
      ResultSet rs = null;
      ResultSet rs1 = null;
      ResultSet rs2 = null;
      try {
         Class.forName("org.apache.derby.jdbc.ClientDriver");
         System.out.println("Driver registered");
         conn =
DriverManager.getConnection("jdbc:derby://localhost:1527/sample"
,"app","app");
         System.out.println("Connection established");
         stmt = conn.createStatement();



         String query1 = "select * from Person";
         rs = stmt.executeQuery(query1);
         Person p[]=new Person[26];
         ArrayList<Person> vacc=new ArrayList<>();
         int i=1;
         while (rs.next()) {
            int aid = rs.getInt(1);
            String name = rs.getString(2);
            int age = rs.getInt(3);
            boolean isComorbid = rs.getBoolean(4);
            p[i]=new Person(aid,name,age,isComorbid);
            int vaccinate=p[i].vaccine(age, isComorbid);
            if(vaccinate==1){
                vacc.add(p[i]);
            }
            i++;
        }
        String query2 = "select * from Employee order by yoe
desc,salary desc,eid asc";
        rs1 = stmt.executeQuery(query2);
        ArrayList<Employee> emps=new ArrayList<>();
        Employee[] empp=new Employee[11];
        int k=1;
        while (rs1.next()) {
            int aid = rs1.getInt(1);
            int eeid = rs1.getInt(2);
            int salary=rs1.getInt(3);
            int yoe=rs1.getInt(4);
            System.out.println("Employee AdhaarId :
"+aid+"\n"+"Employee Id : "+eeid+"\n"+"Salary :
"+salary+"\n"+"Years of Experience : "+yoe+"\n");
            empp[k]=new Employee(aid,eeid,salary,yoe);
            emps.add(new Employee(aid,eeid,salary,yoe));
            k++;
        }

        k=1;
        String query3 = "select fms.eid, fms.fmaid,
emp.yoe,emp.salary from FamilyMembers fms inner join Employee
emp on emp.eid=fms.eid order by emp.yoe desc,emp.salary
desc,emp.eid asc";
        rs2 = stmt.executeQuery(query3);
        int m=1;
        List<Person> fms=new ArrayList<>();
        while(rs2.next()){
                int eeid=rs2.getInt(1);
                int aaid=rs2.getInt(2);
                int yoe=rs2.getInt(3);
                int salary=rs2.getInt(4);
                fms.add(new Person(eeid,aaid,yoe,salary));
        }

        Iterator<Employee> empiterator=emps.iterator();
        Iterator<Person> personiterator=vacc.iterator();
        while(empiterator.hasNext()){
            Employee empl=empiterator.next();
            while(personiterator.hasNext()){
                Person per=personiterator.next();
                if(per.aid==empl.aid){
                    personiterator.remove();
                }
            }
        }
        ArrayList<Person> vaccinate = new ArrayList<>();
        System.out.println("List of Family Memebers of Empolyee for
Vaccination : ");
        System.out.println();
        for (Iterator<Person> periterator = vacc.iterator();
periterator.hasNext();) {
            Person per = periterator.next();
            Person perr=new
Person(per.aid,per.name,per.age,per.isComorbid );
                for (Iterator<Person> perite = fms.iterator();
perite.hasNext();) {
                    Person fmid =perite.next();
                    if(per.aid==fmid.aid){
                        vaccinate.add(new
Person(fmid.eid,perr.aid,perr.name,perr.age,perr.isComorbid,fmid.y
oe,fmid.salary));
                    }
                }
        }
```

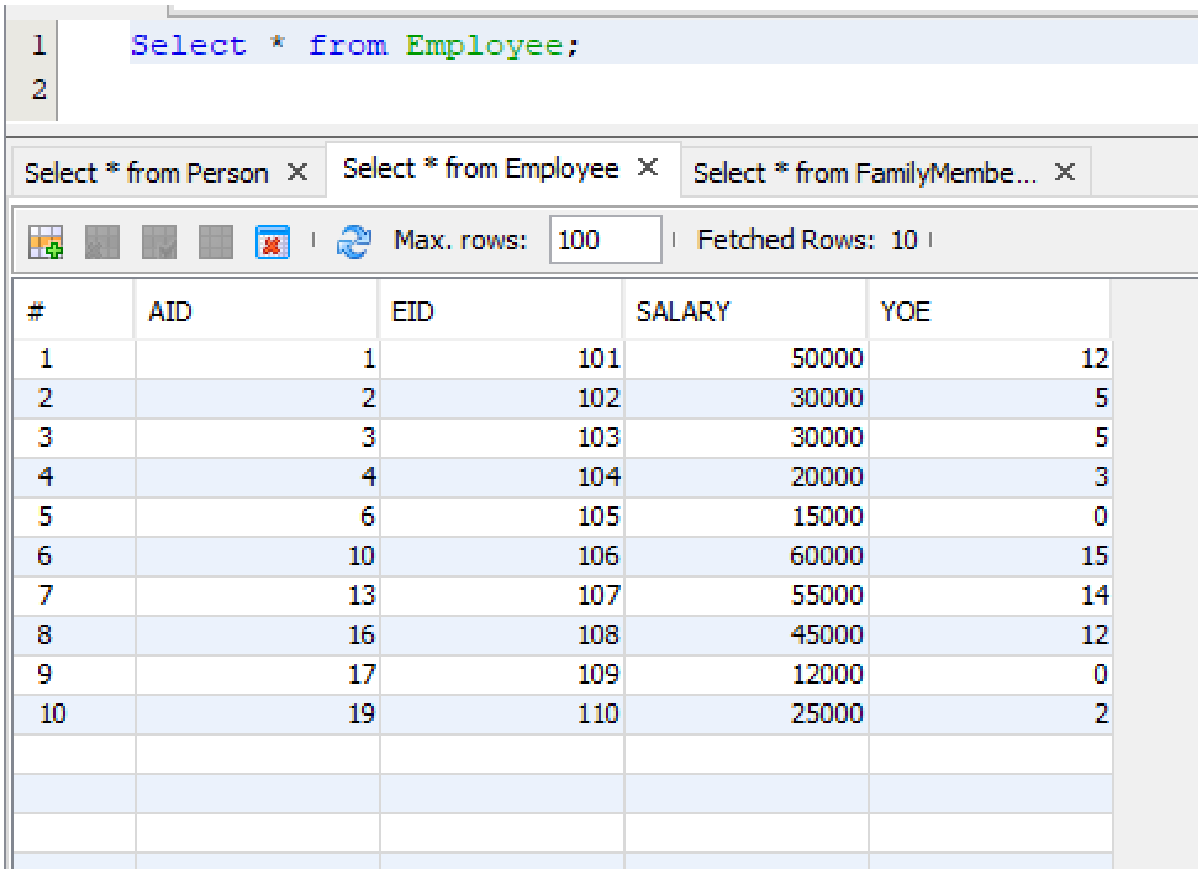
```
            Collections.sort(vaccinate,new multipleSort(new
     yoeSort().reversed(),new salarySort().reversed(),new eidSort()));

            for(int x=0;x<vaccinate.size();x++) {
                        System.out.println(vaccinate.get(x));
                        System.out.println();
                    }
        } catch (ClassNotFoundException ex) {
            System.out.println(ex);
        } catch (SQLException ex) {
            System.out.println(ex);
        } finally {
            try {
                if (rs2 != null) {
                    rs2.close();
                }
                if (rs1 != null) {
                    rs1.close();
                }
                if (rs != null) {
                    rs.close();
                }
                if (stmt != null) {
                    stmt.close();
                }
                if (conn != null) {
                    conn.close();
                }
            } catch (SQLException ex) {
                System.out.println("Exception occured while releasing
resources");
            }
        }
    }
}
```

Output:



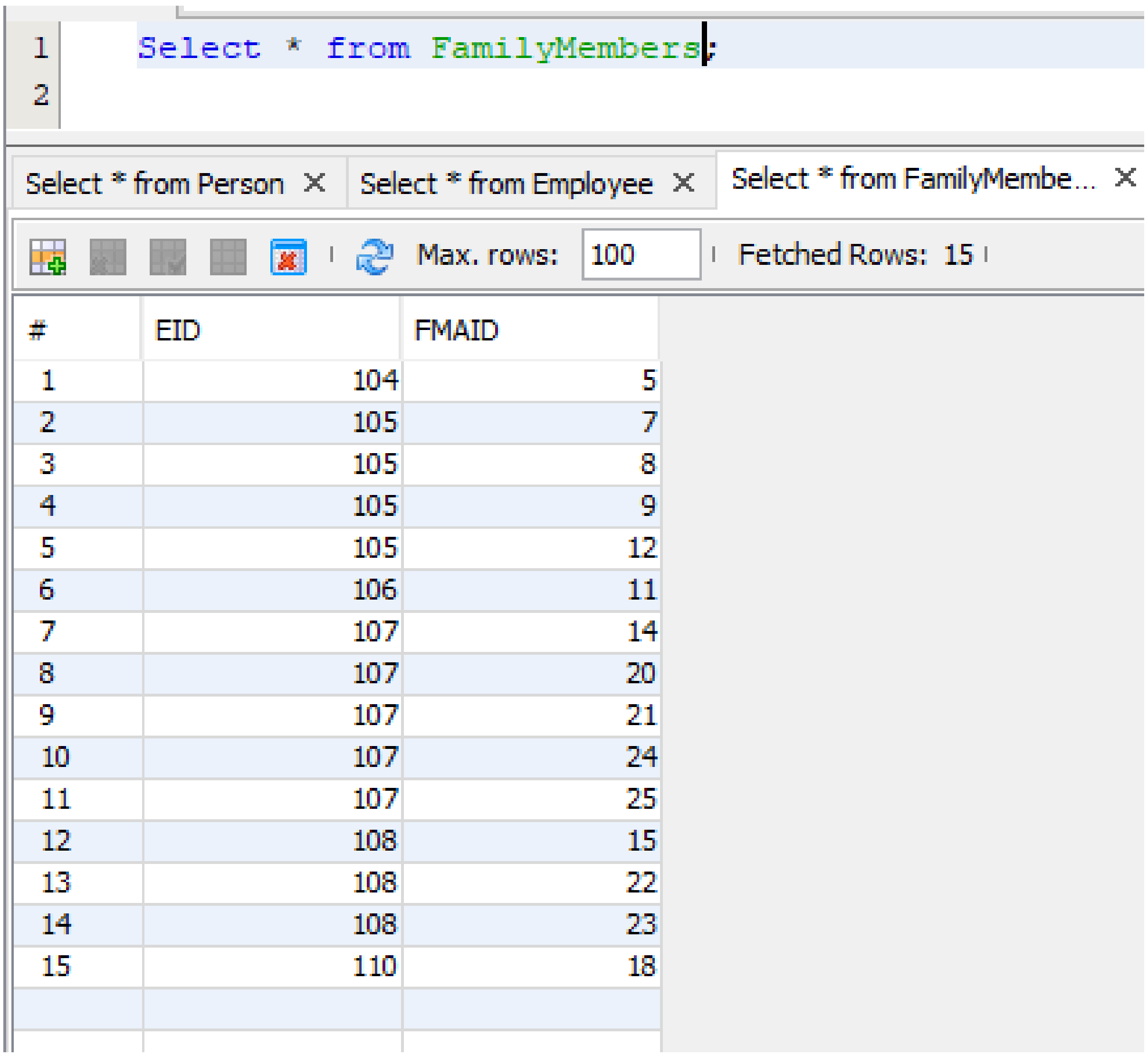
```
Select * from Person;
```

#	AID	NAME	AGE	ISCOMORBID
1	1	aaa	32	☐
2	2	Sheldon	32	☐
3	3	Amy	32	☐
4	4	Lenard	30	✓
5	5	Penny	31	☐
6	6	Harward	33	✓
7	7	Bernadett	32	☐
8	8	Halley	5	☐
9	9	Michael	3	☐
10	10	Rajesh	32	☐
11	11	Anu	32	☐
12	12	Stuart	49	✓
13	13	Ross	36	✓
14	14	Rachel	37	☐
15	15	Monica	31	✓
16	16	Chandler	35	☐
17	17	Joey	33	☐
18	18	Phoebe	30	☐
19	19	Mike	32	✓
20	20	Ben	6	☐
21	21	Emma	3	☐
22	22	Jack	2	☐
23	23	Erica	2	☐
24	24	Jack	72	☐
25	25	Judy	70	✓



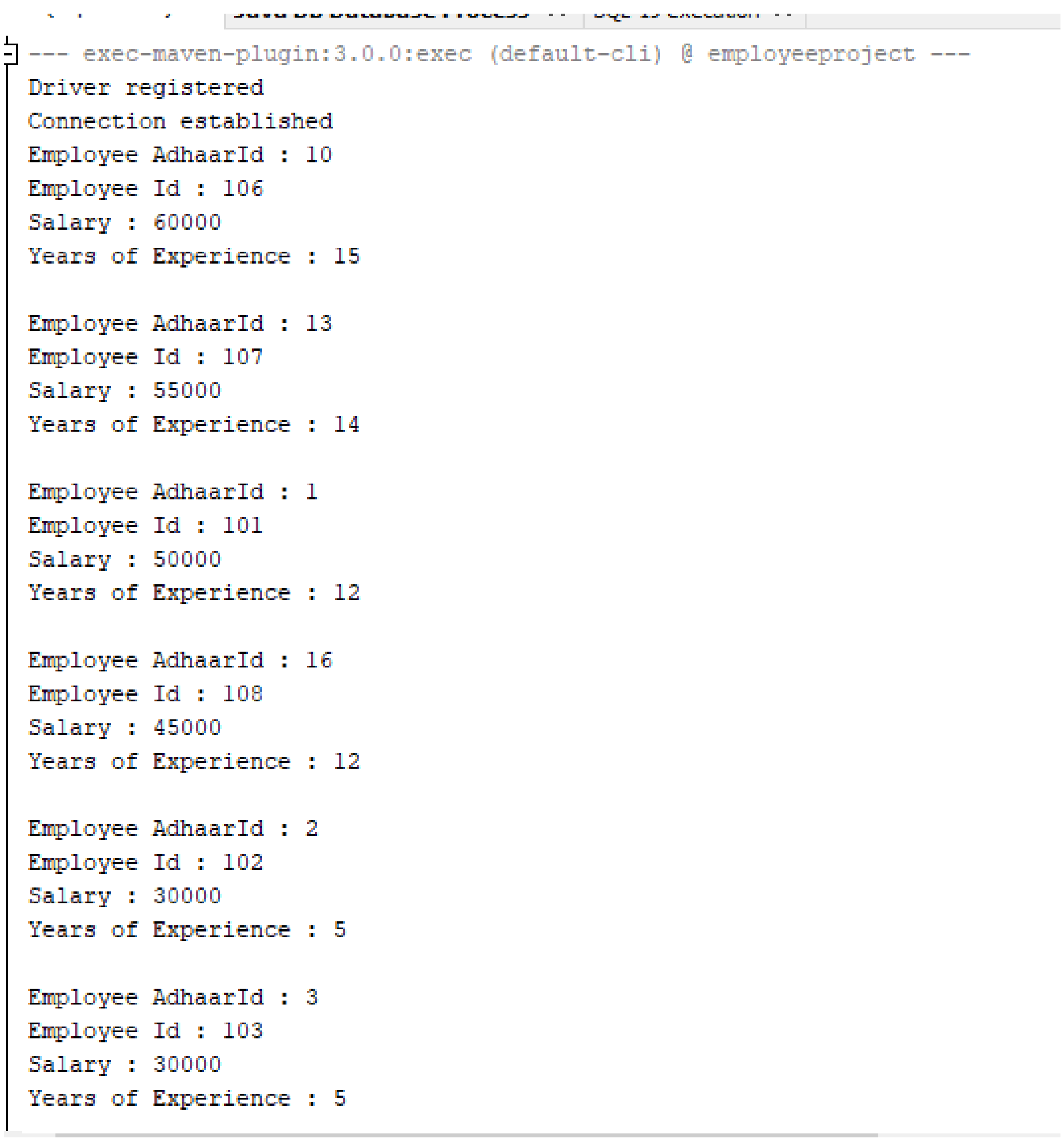
```
Select * from Employee;
```

#	AID	EID	SALARY	YOE
1	1	101	50000	12
2	2	102	30000	5
3	3	103	30000	5
4	4	104	20000	3
5	6	105	15000	0
6	10	106	60000	15
7	13	107	55000	14
8	16	108	45000	12
9	17	109	12000	0
10	19	110	25000	2

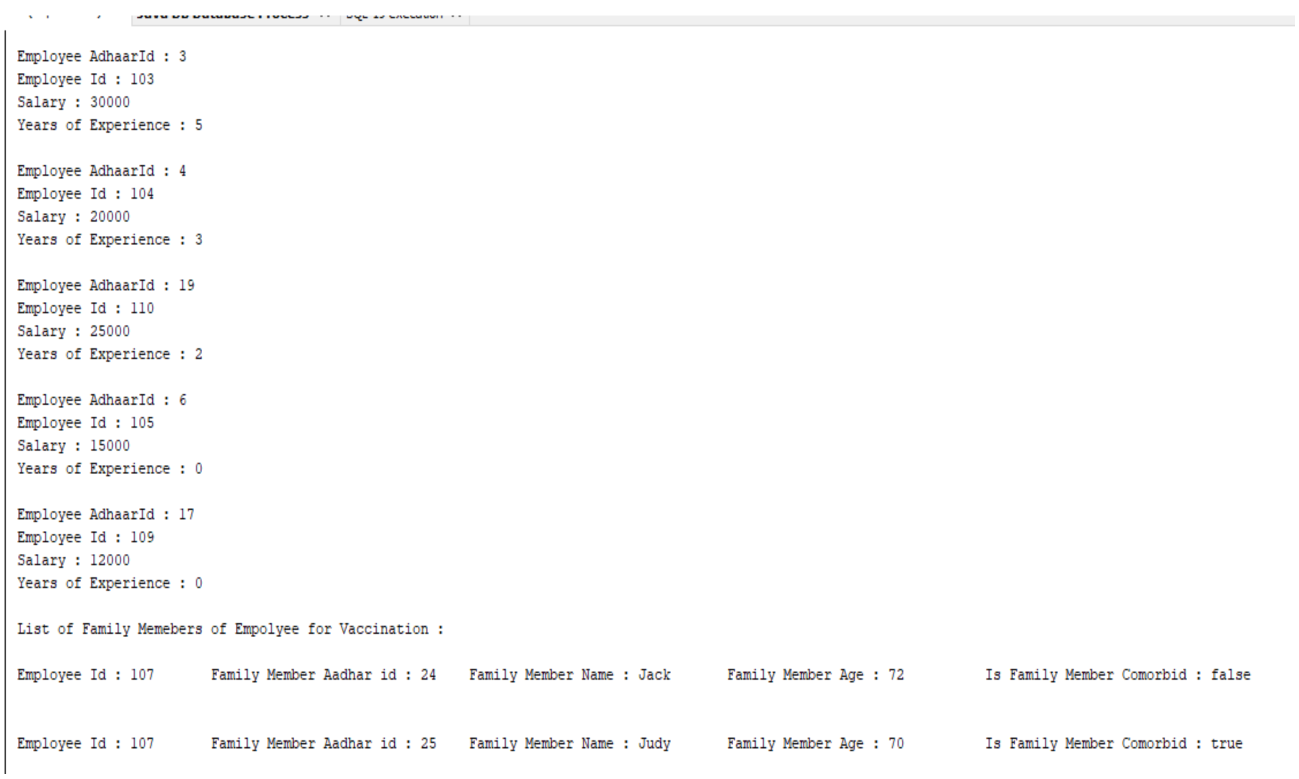


```
Select * from FamilyMembers;
```

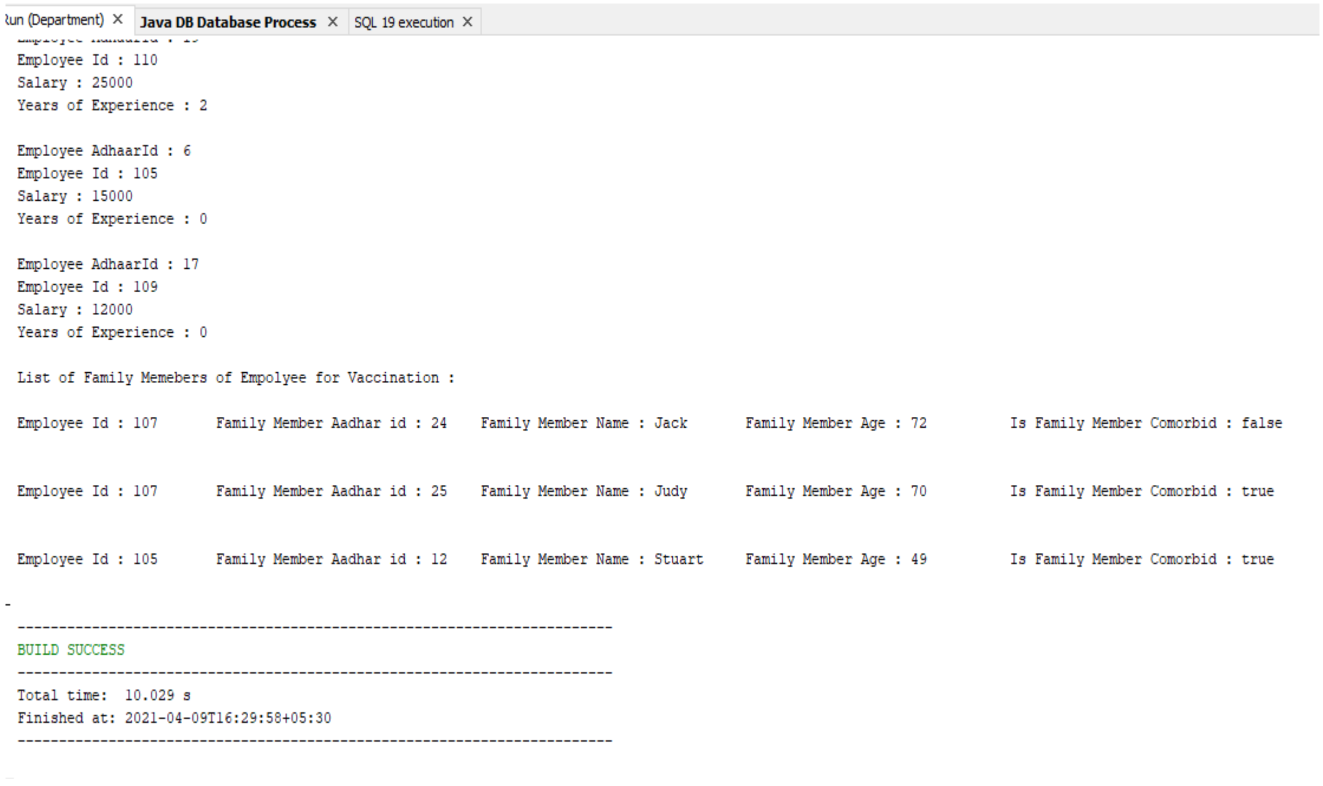
#	EID	FMAID
1	104	5
2	105	7
3	105	8
4	105	9
5	105	12
6	106	11
7	107	14
8	107	20
9	107	21
10	107	24
11	107	25
12	108	15
13	108	22
14	108	23
15	110	18



```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ employeeproject ---
Driver registered
Connection established
Employee AdhaarId : 10
Employee Id : 106
Salary : 60000
Years of Experience : 15

Employee AdhaarId : 13
Employee Id : 107
Salary : 55000
Years of Experience : 14

Employee AdhaarId : 1
Employee Id : 101
Salary : 50000
Years of Experience : 12

Employee AdhaarId : 16
Employee Id : 108
Salary : 45000
Years of Experience : 12

Employee AdhaarId : 2
Employee Id : 102
Salary : 30000
Years of Experience : 5

Employee AdhaarId : 3
Employee Id : 103
Salary : 30000
Years of Experience : 5
```



```
Employee AdhaarId : 3
Employee Id : 103
Salary : 30000
Years of Experience : 5

Employee AdhaarId : 4
Employee Id : 104
Salary : 20000
Years of Experience : 3

Employee AdhaarId : 19
Employee Id : 110
Salary : 25000
Years of Experience : 2

Employee AdhaarId : 6
Employee Id : 105
Salary : 15000
Years of Experience : 0

Employee AdhaarId : 17
Employee Id : 109
Salary : 12000
Years of Experience : 0

List of Family Memebers of Empolyee for Vaccination :

Employee Id : 107       Family Member Aadhar id : 24    Family Member Name : Jack       Family Member Age : 72          Is Family Member Comorbid : false


Employee Id : 107       Family Member Aadhar id : 25    Family Member Name : Judy       Family Member Age : 70          Is Family Member Comorbid : true
```



```
Employee Id : 110
Salary : 25000
Years of Experience : 2

Employee AdhaarId : 6
Employee Id : 105
Salary : 15000
Years of Experience : 0

Employee AdhaarId : 17
Employee Id : 109
Salary : 12000
Years of Experience : 0

List of Family Memebers of Empolyee for Vaccination :

Employee Id : 107       Family Member Aadhar id : 24    Family Member Name : Jack       Family Member Age : 72          Is Family Member Comorbid : false


Employee Id : 107       Family Member Aadhar id : 25    Family Member Name : Judy       Family Member Age : 70          Is Family Member Comorbid : true


Employee Id : 105       Family Member Aadhar id : 12    Family Member Name : Stuart     Family Member Age : 49          Is Family Member Comorbid : true

------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  10.029 s
Finished at: 2021-04-09T16:29:58+05:30
------------------------------------------------------------------------
```