INDIVIDUAL PROJECT: PAN

COURSE:CS/DSA-4513 -DATABASE MANAGEMENT

SECTION:001

SEMESTER: FALL 2020

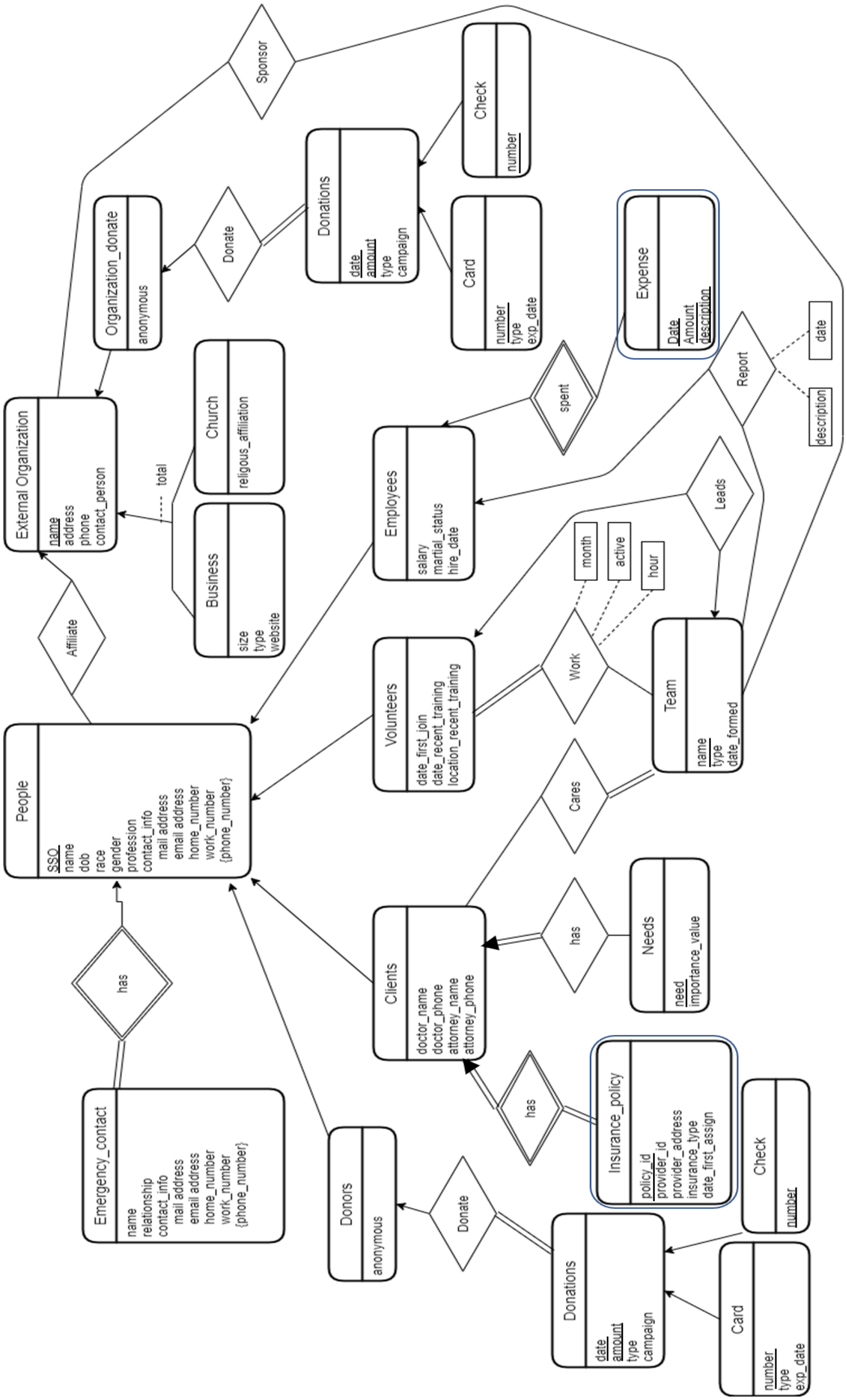
INSTRUCTOR: DR. LE GRUENWALD

STUDENT: CHUONG LE

EMAIL: chuongl@ou.edu

Task 1:

ER Diagram:



Relational Database

People (SSN, p_name, dob, race, gender, profession, m_address, e_address, h_number, w_number, c_number, on_list)

Emergency (SSN, e_name, relationship, email_address, mailing_address, home _number, work _number, cell _number)

Client (SSN, d_name, d_phone, a_name, a_phone, date_first_assigned)

Needs (SSN, needs, importance_value)

Insurance_Policy (SSN, pol_id, pro_id, pro_address, i_type)

Team (t_name, t_type, date_formed)

Cares (SSN, t_name, active)

Volunteers (SSN, date_first_join, date_recent_train, location_recent_train)

Serves (SSN, t_name, months, hour, active)

Leads (SSN, t_name)

Employee (SSN, salary, marital_status, hired_date)

Reporting (SSN, t_name, r_date, r_description)

Expenses (SSN, e_date, amount, e_description)

Donors (SSN, anonymous)

Donor_Donate (SSN, d_date, d_amount, type, d_campaign)

Donor_Donate_Check (SSN, d_date, d_amount, d_check_num)

Donors_Donate_Card (ssn, d_date, d_amount, d_card_number, d_card_type, d_card_exp)

Affiliates (SSN, org_name)

External_Organization (org_name, org_mailing, org_phone, contact_person)

Church (org_name, religious_affiliation)

Business (org_name, b_type, size, website)

Sponsor (org_name, t_name)

Org_Donate (org_name, org_date, org_amount, type, org_campaign).

Org_Donate_Check (org_name, date, amount, Check_Number).

Org_Donates_Card (org_name, date, amount, Card_Number, Card_Type,

Expiry_Date).

Organization_Donors (org_name, anonymous)

Task 2: Data Dictionary

Table	Column	Datatype	Data Size	Constraint
People	SSN	INT	4	PK, NOT NULL
	P_name	VARCHAR	64	
	Dob	INT	4	
	Race	VARCHAR	64	
	Profession	VARCHAR	64	
	M_address	VARCHAR	64	
	E_address	VARCHAR	64	
	H_number	INT	4	
	W_number	INT	4	
	C_number	INT	4	
	On_list	VARCHAR	64	
Emergency	SSN	INT	4	PK, FK People
	E_name	VARCHAR	64	
	Relationship	VARCHAR	64	
	Email_address	VARCHAR	64	
	Mailing Address	VARCHAR	64	
	Home_number	INT	4	
	Work_nuber	INT	64	
	Cell_number	INT	64	PK
Client	SSN	INT	4	PK, FK People
	d_name	VARCHAR	64	
	d_phone	INT	4	

	a_name	VARCHAR	64	
	a_phone	INT	4	
	date_first_assigned	INT	4	
Needs	SSN			PK, FK Client
	needs			PK
	importance_value			
Insurance_Policy	SSN	INT	4	PK, FK Client
	pol_id	INT	4	PK
	pro_id	INT	4	
	pro_address	VARCHAR	64	
	i_type	VARCHAR	64	
Team	t_name	VARCHAR	64	PK
	t_type	VARCHAR	64	
	date_formed	VARCHAR	64	
Cares	SSN	INT	4	PK, FK Client
	t_name	VARCHAR	64	PK, FK Team
	active	VARCHAR	64	
Serves	SSN	INT	4	PK, FK Volunteer
	t_name	VARCHAR	64	PK, FK Team
	months	INT	4	PK
	hour	INT	4	
	active	VARCHAR	64	
Volunteers	SSN	INT	4	PK, FK People
	date_first_join	INT	4	

	date_recent_train	INT	4	
	location_recent_train	VARCHAR	64	
Employee	SSN	INT	4	PK, FK People
	salary	INT	4	
	marital Status	VARCHAR	64	
	hired_date	INT	4	
Leads	SSN	INT	4	PK, FK Volunteer
	t_name	VARCHAR	64	FK Team
Expense	SSN	INT	4	PK, FK Employee
	e_date	INT	4	PK
	amount	INT	4	PK
	e_description	VARCHAR	64	PK
Reporting	SSN	INT	4	PK,FK Employee
	t_name	VARCHAR	64	PK
	r_date	INT	4	
	r_description	VARCHAR	64	
Donor	SSN	INT	4	PK,FK People
	anonymous	VARCHAR	64	
Donor_Donate	SSN	INT	4	PK, FK Donor
	d_date	INT	4	PK
	d_amount	INT	4	PK
	d_type	VARCHAR	64	
	d_campaign	VARCHAR	64	
Donor_Donate_	SSN	INT	4	PK, FK

Check				Donor_Donate
	d_date	INT	4	PK, FK Donor_Donate
	d_amount	INT	4	PK, FK Donor_Donate
	d_check_num	INT	4	PK
Donor_Donate_Card	SSN	INT	4	PK, FK Donor_Donate
	d_date	INT	4	PK, FK Donor_Donate
	d_amount	INT	4	PK. FK Donor_Donate
	d_card_num	INT	4	PK
	d_card_type	VARCHAR	64	
	d_card_exp	INT	4	
External Organization	org_name	VARCHAR	64	PK
	org_mailing	VARCHAR	64	
	org_phone	INT	4	
	contact_people	VARCHAR	64	
Affiliates	SSN	INT	4	PK, FK People
	org_name	VARCHAR	64	PK, FK External_Organization
Church	org_name	VARCHAR	64	PK, FK External_Organization
	religous_affiliation	VARCHAR	64	
Business	org_name	VARCHAR	64	PK, FK External_Organization

	b_type	VARCHAR	64	
	size	VARCHAR	64	
	website	VARCHAR	64	
Sponsor	org_name	VARCHAR	64	PK, FK External_Organization
	t_name	VARCHAR	64	PK, FK Team
Organization_Donor	org_name	VARCHAR	64	PK, FK External_Organization
	anonymous	VARCHAR	64	
Organization_Donate	org-name	VARCHAR	64	PK, FK Organization_Donor
	donate_date	INT	4	PK
	donate_amount	INT	4	PK
	donate_type	VARCHAR	64	
	donate_campaign	VARCHAR	64	
Organization_Donate_Check	org_name	VARCHAR	64	PK, FK Organization_Donate
	donate_date	INT	4	PK, FK Organization_Donate
	donate_amount	INT	4	PK, FK Organization_Donate
	donate_check_num	INT	4	PK
Organization_Donate_Card	org-name	VARCHAR	64	PK, FK Organization_Donate
	donate_date	INT	4	PK, FK

				Organization_Donate
	donate_amount	INT	4	PK,FK Organization_Donate
	donate_card_num	INT	4	PK
	donate_card_type	VARCHAR	64	
	donate_card_exp	INT	4	

Task 3:

Storage Structures:

Table	**Query**	**Search Key**	**Frequency**	**Structure**
People	20. Insert 13. Random Search 14. Random Search	 SSN SSN	 1/Week 1/Week	 Dynamic hashing with SSN Dynamic hashing with SSN
Emergency				Heap
Client	2. Insert 10. Random Search 13. Random Search 17. Delete	 SSN SSN SSN	1/Week 1/Week 1/Week 4/Year	 Dynamic hashing with SSN Dynamic hashing with SSN Dynamic hashing with SSN
Needs	2. Insert 17. Random Search 17. Random Search	 SSN need	1/Week 4/Year 4/Year	 Dynamic Hashing with Needs Dynamic Hashing with Needs
Insurance_policy	7. Insert 17 Random	 i_type	1/Week 4/Year	Dynamic hashing with i_type
Team	1. Insert 13 Random Search 15. Random Search	 t_name t_name	1/Month 1/Week 1/Month	 Index sequential file
Cares	2. Insert 12. Random Search 13. Random Search	 t_name t_name	1/Week 4/Year 1/Week	 Dynamic hashing with t_name Dynamic hashing with t_name
Serves	3. Insert 4. Insert 12. Random	 SSN	2/Month 30/Month 4/Year	 Dynamic hashing

				with t_name
Volunteers	3. Insert 12. Random Search	SSN	2/Month 4/Year	Dynamic hashing with SSN
Employee	5. Insert 6. Random Search 14. Random Search 16. Random Search	SSN SSN SSN	1/Year 1/Day 1/Week 1/Year	Dynamic hashing with SSN Dynamic hashing with SSN Dynamic hashing with SSN
Leads	3. Insert		2/Month	Heap
Expense	6. Insert 11. Range Search	date_formed	2/Day 1/Month	Index Sequential File
Reporting	5. Insert 16. Random Search	SSN	1/Year 1/Year	Dynamic Hashing with SSN
Donor	8. Insert 14. Random	SSN	1/Day 1/Week	Dynamic hashing with SSN
Donor_Donate	8. Insert 14. Random Search	SSN	1/Day 1/Week	Dynamic hashing with SSN
Donor_Donate_Check	8. Insert		1/Day	Heap
Donor_Donate_Card	8. Insert		1/Day	Heap
External_Organization	7. Insert 9. Insert		2/Week 1/Day	Heap
Affiliates				Heap
Church				Heap
Business				Heap
Sponsor	7. Insert		2/Week	

	13. Range Search	o_name	1/Week	Dynamic Hashing with o_name
Organization_Donor	9. Insert		1/Day	Heap
Organization_Donate	9. Insert		1/Day	Heap
Organization_Donate_Check	9. Insert		1/Day	Heap
Organization_Donate_Card	9. Insert		1/Day	Heap

Alternate Storage Structure:

Some of the secondary key index is:

```
CREATE INDEX expense_amount_index
ON Expense (name)
```

```
CREATE INDEX team_date_index
ON Team(date_formed)
```

An alternative would be to use the B-Tree for indexing.

Task 4: SQL code:

```
DROP TABLE IF EXISTS Organization_Donate_Card
DROP TABLE IF EXISTS Organization_Donate_Check
DROP TABLE IF EXISTS Organization_Donate
DROP TABLE IF EXISTS Organization_Donor
DROP TABLE IF EXISTS Sponsor
DROP TABLE IF EXISTS Business
DROP TABLE IF EXISTS Church
DROP TABLE IF EXISTS Affiliates
DROP TABLE IF EXISTS External_Organization
DROP TABLE IF EXISTS Donor_Donate_Card
DROP TABLE IF EXISTS Donor_Donate_Check
DROP TABLE IF EXISTS Donor_Donate
DROP TABLE IF EXISTS Donor
DROP TABLE IF EXISTS Reporting
DROP TABLE IF EXISTS Expenses
DROP TABLE IF EXISTS Leads
DROP TABLE IF EXISTS Employee
DROP TABLE IF EXISTS Volunteers
DROP TABLE IF EXISTS Serves
DROP TABLE IF EXISTS Cares
DROP TABLE IF EXISTS Team
DROP TABLE IF EXISTS Insurance_Policy
DROP TABLE IF EXISTS Needs
DROP TABLE IF EXISTS Client
DROP TABLE IF EXISTS Emergency
DROP TABLE IF EXISTS People



CREATE Table People(
    SSN INT PRIMARY KEY NOT NULL,
    P_name VARCHAR(64) ,
    Dob INT,
    Race VARCHAR(64),
    Gender VARCHAR(64) ,
    Profession VARCHAR(64) ,
    M_address VARCHAR(64) ,
    E_address VARCHAR(30) ,
    H_number INT ,
    W_number INT ,
    C_number INT ,
    On_list VARCHAR(64)
)

CREATE TABLE Emergency (
    SSN INT NOT NULL,
    E_name VARCHAR(64),
    Relationship VARCHAR(64),
    Email_address VARCHAR(30) NOT NULL,
    Mailing_address VARCHAR(64) NOT NULL,
    Home_number INT NOT NULL,
    Work_number INT NOT NULL,
    Cell_number INT NOT NULL,
    PRIMARY KEY (SSN, Cell_number),
    FOREIGN KEY (SSN) REFERENCES People(SSN) ON DELETE CASCADE
)

CREATE TABLE Client(
    SSN INT PRIMARY KEY,
    d_name VARCHAR(64),
    d_phone INT,
    a_name VARCHAR(64),
    a_phone INT,
    date_first_assigned INT,
    FOREIGN KEY (SSN) REFERENCES People(SSN) ON DELETE CASCADE
)

CREATE TABLE Needs(
    SSN INT,
    needs VARCHAR(20),
    importance_value INT,
    PRIMARY KEY (SSN,Needs),
    FOREIGN KEY (SSN) REFERENCES Client(SSN) ON DELETE CASCADE
)

CREATE TABLE Insurance_Policy(
    SSN INT,
    pol_id VARCHAR(64) PRIMARY KEY,
    pro_id VARCHAR(64),
    pro_address VARCHAR(64),
    i_type VARCHAR(20),
    FOREIGN KEY (SSN) REFERENCES Client(SSN) ON DELETE CASCADE,
)

CREATE TABLE Team(
    t_name VARCHAR(64) PRIMARY KEY,
    t_type VARCHAR(64),
    date_formed VARCHAR(64),
)

CREATE TABLE Cares(
    SSN INT,
    t_name VARCHAR(64),
    active VARCHAR(64),
    PRIMARY KEY (SSN, t_name),
    FOREIGN KEY (SSN) REFERENCES Client(SSN) ON DELETE CASCADE,
    FOREIGN KEY (t_name) REFERENCES Team(t_name) ON DELETE CASCADE
)

CREATE TABLE Serves(
    SSN INT,
    t_name VARCHAR(64),
    months INT,
    hour INT,
    active VARCHAR(64),
    PRIMARY KEY (SSN, t_name, months),
    FOREIGN KEY (SSN) REFERENCES People(SSN) ON DELETE CASCADE,
    FOREIGN KEY (t_name) REFERENCES Team(t_name) ON DELETE CASCADE
)

CREATE TABLE Volunteers(
    SSN INT PRIMARY KEY,
    date_first_join INT,
    date_recent_train INT,
    location_recent_train VARCHAR(64),
    FOREIGN KEY (SSN) REFERENCES People(SSN) ON DELETE CASCADE
)

CREATE TABLE Employee(
    SSN INT PRIMARY KEY,
    salary INT,
    marital_Status VARCHAR(64),
    hired_date INT,
    FOREIGN KEY (SSN) REFERENCES People(SSN) ON DELETE CASCADE
)

CREATE TABLE Leads(
    SSN INT PRIMARY KEY,
    t_name VARCHAR(64),
    FOREIGN KEY (SSN) REFERENCES People(SSN) ON DELETE CASCADE,
    FOREIGN KEY (t_name) REFERENCES Team(t_name) ON DELETE CASCADE
)

CREATE TABLE Expenses(
    SSN INT,
    e_date INT,
    amount INT,
    e_description varchar(64),
    PRIMARY KEY (SSN, e_date, amount, e_description),
    FOREIGN KEY (SSN) REFERENCES Employee(SSN) ON DELETE CASCADE
```

```
)
CREATE TABLE Reporting(
    SSN INT,
    t_name VARCHAR(64),
    r_date INT,
    r_description VARCHAR(64),
    PRIMARY KEY (t_name,SSN),
    FOREIGN KEY (SSN) REFERENCES Employee(SSN) ON DELETE CASCADE,
    FOREIGN KEY (t_name) REFERENCES Team(t_name) ON DELETE CASCADE
)



CREATE TABLE Donor(
    SSN INT PRIMARY KEY,
    anonymous VARCHAR(64),
    FOREIGN KEY (SSN) REFERENCES People(SSN) ON DELETE CASCADE
)

CREATE TABLE Donor_Donate(
    SSN INT,
    d_date INT,
    d_amount INT,
    d_type VARCHAR(64),
    d_campaign VARCHAR(64),
    PRIMARY KEY (SSN, d_date ,d_amount),
    FOREIGN KEY (SSN) REFERENCES Donor(SSN) ON DELETE CASCADE
)

CREATE TABLE Donor_Donate_Check(
    SSN INT,
    d_date INT,
    d_amount INT,
    d_check_num VARCHAR(64),
    PRIMARY KEY (SSN,d_date, d_amount, d_check_num),
    FOREIGN KEY (SSN, d_date, d_amount) REFERENCES Donor_Donate(SSN, d_date, d_amount) ON
DELETE CASCADE
)

CREATE TABLE Donor_Donate_Card(
    SSN INT,
    d_date INT,
    d_amount INT,
    d_card_num INT,
    d_card_type VARCHAR(64),
    d_card_exp INT,
    PRIMARY KEY (SSN,d_date, d_amount, d_card_num),
    FOREIGN KEY (SSN, d_date, d_amount) REFERENCES Donor_Donate(SSN, d_date, d_amount) ON
DELETE CASCADE
)

CREATE TABLE External_Organization(
    org_name VARCHAR(64) PRIMARY KEY,
    org_mailing VARCHAR(64),
    org_phone INT,
    contact_People VARCHAR(64)
)
CREATE TABLE Affiliates(
    SSN INT PRIMARY KEY,
    org_name VARCHAR(64),
    FOREIGN KEY (SSN) REFERENCES People(SSN) ON DELETE CASCADE,
    FOREIGN KEY(org_name) REFERENCES External_Organization(org_name) ON DELETE CASCADE
)

CREATE TABLE Church(
    org_name VARCHAR(64) PRIMARY KEY,
    religious_affiliation VARCHAR(64),
    FOREIGN KEY (org_name) REFERENCES External_Organization(org_name) ON DELETE CASCADE
)

CREATE TABLE Business(
    org_name VARCHAR(64) PRIMARY KEY,
    b_type VARCHAR(64),
    size VARCHAR(64),
    website VARCHAR(64),
    FOREIGN KEY (org_name) REFERENCES External_Organization(org_name) ON DELETE CASCADE
)

CREATE TABLE Sponsor(
    org_name VARCHAR(64),
    t_name VARCHAR(64),
    PRIMARY KEY (org_name, t_name),
    FOREIGN KEY (org_name) REFERENCES External_Organization(org_name) ON DELETE CASCADE,
    FOREIGN KEY (t_name) REFERENCES Team(t_name) ON DELETE CASCADE
)

CREATE TABLE Organization_Donor(
    org_name VARCHAR(64) PRIMARY KEY,
    anonymous VARCHAR(64),
    FOREIGN KEY (org_name) REFERENCES External_Organization(org_name) ON DELETE CASCADE
)
```

```
CREATE TABLE Organization_Donate(
    org_name VARCHAR(64),
    donate_date INT,
    donate_amount INT,
    donate_type VARCHAR(64),
    donate_campaign varchar(64),
    PRIMARY KEY (org_name, donate_date, donate_amount),
    FOREIGN KEY (org_name) REFERENCES Organization_Donor(org_name) ON DELETE CASCADE
)

CREATE TABLE Organization_Donate_Check(
    org_name VARCHAR(64),
    donate_date INT,
    donate_amount INT,
    donate_check_num VARCHAR(64),
    PRIMARY KEY(org_name, donate_date, donate_amount, donate_check_num),
    FOREIGN KEY (org_name, donate_date, donate_amount) REFERENCES
Organization_Donate(org_name, donate_date, donate_amount) ON DELETE CASCADE
)

CREATE TABLE Organization_Donate_Card(
    org_name VARCHAR(64),
    donate_date INT,
    donate_amount INT,
    donate_card_num INT,
    donate_card_type VARCHAR(64),
    donate_card_exp INT,
    PRIMARY KEY(org_name, donate_date, donate_amount, donate_card_num),
    FOREIGN KEY (org_name, donate_date, donate_amount) REFERENCES
Organization_Donate(org_name, donate_date, donate_amount) ON DELETE CASCADE
)

CREATE INDEX expense_amount_index
ON Expenses(amount)

CREATE INDEX team_date_index
ON Team(date_formed)
```

Task 5: Java Code

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.Connection;
import java.sql.Statement;
import java.util.Scanner;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

public class sample {

   // Database credentials
   final static String HOSTNAME = "<username>-sql-server.database.windows.net";
   final static String DBNAME = "cs-dsa-4513-sql-db";
   final static String USERNAME = "<username>";
   final static String PASSWORD = "<password>";
   static Connection conn;

   // Database connection string
   final static String URL =
String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;trust
ServerCertificate=false;hostNameInCertificate=*.database.windows.net;loginTimeout=30;",
         HOSTNAME, DBNAME, USERNAME, PASSWORD);
   public static Scanner sc = new Scanner(System.in); // Scanner is used to collect the user input


   final static String Query1 = "INSERT INTO Team(t_name, t_type, date_formed) VALUES
(?,?,?)";
   final static String Query2 = "INSERT INTO CLIENT (SSN, d_name, d_phone, a_name,
a_phone, date_first_assigned) VALUES (?,?,?,?,?,?)";
   final static String Query2_care = "INSERT INTO Cares (SSN, t_name, active) VALUES
(?,?,?)";
   final static String Query2_insurance = "INSERT INTO Insurance_Policy (SSN, pol_id,
pro_id, pro_address, i_type) VALUES (?,?,?,?,?)";
   final static String Query2_need = "INSERT INTO Needs (SSN, needs, importance_value)
VALUES (?,?,?)";
```

```
   final static String Query3 = "INSERT INTO Volunteers (SSN, date_first_join,
date_recent_train, location_recent_train) VALUES (?,?,?,?)";
   final static String Query34_serve = "INSERT INTO Serves (SSN, t_name, months, hour,
active) VALUES (?,?,?,?,?)";
   final static String Query34_active = "INSERT INTO Active (SSN, t_name, active) VALUES
(?,?,?)"; /*  might not need */
   final static String Query5 = "INSERT INTO Employee (SSN,salary,marital_Status,hired_date)
VALUES (?,?,?,?)";
   final static String Query5_report = "INSERT INTO Reporting
(SSN,t_name,r_date,r_description) VALUES (?,?,?,?)";
   final static String Query6 = "INSERT INTO Expenses (SSN,e_date,amount,e_description)
VALUES (?,?,?,?)";
   final static String Query7 = "INSERT INTO External_Organization
(org_name,org_mailing,org_phone,contact_People) VALUES (?,?,?,?)";
   final static String Query7_sponsor = "INSERT INTO Sponsor (org_name, t_name) VALUES
(?,?)";
   final static String Query8 = "INSERT INTO Donor (SSN, anonymous) VALUES (?,?)";
   final static String Query8_Donate = "INSERT INTO Donor_Donate (SSN, d_date,
d_amount,d_type,d_campaign) VALUES (?,?,?,?,?)";
   final static String Query8_Card = "INSERT INTO Donor_Donate_Card (SSN, d_date,
d_amount, d_card_num, d_card_type, d_card_exp) VALUES (?,?,?,?,?,?)";
   final static String Query8_Check = "INSERT INTO Donor_Donate_Check (SSN, d_date,
d_amount, d_check_num) VALUES (?,?,?,?)";
   final static String Query9 = "INSERT INTO Organization_Donor VALUES (?,?)";
   final static String Query9_donate = "INSERT INTO Organization_Donate VALUES
(?,?,?,?,?)";
   final static String Query9_Card = "INSERT INTO Organization_Donate_Card VALUES
(?,?,?,?,?,?)";
   final static String Query9_Check = "INSERT INTO Organization_Donate_Check VALUES
(?,?,?,?)";

   // User input prompt//
   final static String PROMPT =
                              "Enter you option(1-17):\n" +
                              "1. Enter a new team into the database\n" +
                              "2. Enter a new client into the database and associate him or her
with one or more teams\n" +
                              "3. Enter a new volunteer into the database and associate him or
her with one or more teams\n" +
                            "4. Enter the number of hours a volunteer worked this month for a
particular team\n" +
                            "5. Enter a new employee into the database and associate him or
her with one or more teams\n" +
                            "6. Enter an expense charged by an employee\n" +
                            "7. Enter a new organization and associate it to one or more PAN
teams\n" +
                            "8. Enter a new donor and associate him or her with several
donations.\n" +
                            "9. Enter a new organization and associate it with several
donations\n" +
                            "10. Retrieve the name and phone number of the doctor of a
particular client\n" +
                            "11. Retrieve the total amount of expenses charged by each
employee for a particular period of time.\r\n" +
                            "The list should be sorted by the total amount of expenses\n" +
                            "12. Retrieve the list of volunteers that are members of teams that
support a particular client\n" +
                            "13. Retrieve the names and contact information of the clients that
are supported by teams sponsored by\r\n" +
                            "an organization whose name starts with a letter between B and K.
The client list should be sorted by name\n" +
                            "14. Retrieve the name and total amount donated by Donor that are
also employees. The list should be\r\n" +
                            "sorted by the total amount of the donations, and indicate if each
donor wishes to remain anonymous\n" +
                            "15. Retrieve the names of all teams that were founded after a
particular date(1/month)\n" +
                            "16. Increase the salary by 10% of all employees to whom more
than one team must report\n"+
                            "17. Delete all clients who do not have health insurance and whose
value of importance for transportation\r\n" +
                            "is less than 5\n"+
                            "18. Import\n"+
                            "19. Export\n" +
                            "20. Add a Person\n" +
                            "21. Close";

    public static void main(String[] args) throws SQLException {
        conn = DriverManager.getConnection(URL);
```

```
String option = ""; // Initialize user option selection as nothing
while (!option.equals("21")) { // As user for options until option 4 is selected
  System.out.println(PROMPT); // Print the available options
  option = sc.nextLine(); // Read in the user option selection

  switch (option) { // Switch between different options
    case "1": // Insert a new faculty with first salary calculation method option
      Q1();
      break;
    case "2":
        Q2();
        break;
    case "3":
                          Q3();
                          break;
                    case "4":
                          Q4();
                          break;
                    case "5":
                          Q5();
                          break;
                    case "6":
                          Q6();
                          break;
                    case "7":
                          Q7();
                          break;
                    case "8":
                          Q8();
                          break;
                    case "9":
                          Q9();
                          break;
                    case "10":
                          Q10();
                          break;
                    case "11":
                          Q11();
                          break;
                    case "12":
```

```
                                                        Q12();
                                                        break;
                                            case "13":
                                                        Q13();
                                                        break;
                                            case "14":
                                                        Q14();
                                                        break;
                                            case "15":
                                                        Q15();
                                                        break;
                                            case "16":
                                                        Q16();
                                                        break;
                                            case "17":
                                                        Q17();
                                                        break;
                                            case "18":
                                                        Q18();
                                                        break;
                                            case "19":
                                                        Q19();
                                                        break;
                                            case "20":
                                                        Q20();
                                                        break;
                                            case "21":
                                                        sc.close();
                                                        System.out.println("Later!");
                                                        break;
                                            default:
                                                        System.out.println("Sorry, Unrecognized input");
                                                        break;
        }
    }

    sc.close(); // Close the scanner before exiting the application
}

public static void Q1() throws SQLException
```

```
{

  System.out.println("Please enter team name:");
  String name = sc.nextLine();
  //sc.nextLine();
  System.out.println("Please enter team type:");
  String type = sc.nextLine();

  System.out.println("Please enter date formed: 010112 = Jan 1, 2012");
  String date = sc.nextLine();
  try (final PreparedStatement statement = conn.prepareStatement(Query1))
  {
    // Populate the query template with the data collected from the user
    statement.setString(1, name);
    statement.setString(2, type);
    statement.setString(3, date);

    final int rows_inserted = statement.executeUpdate();
    System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
  }
  displayTeam();

}
/* TO TEST YOU NEED to insert FOREIGN KEY before running*/
public static void Q2() throws SQLException
{
    /* Client */
    System.out.println("Enter client SSN from People table:");
    int SSN = Integer.parseInt(sc.nextLine());

    System.out.println("Enter Doctor's name");
    String d_name = sc.nextLine();

    System.out.println("Enter Doctor's phone number: ex - 4051234567");
    int d_phone = Integer.parseInt(sc.nextLine());

    System.out.println("Enter Attorney's name");
    String a_name = sc.nextLine();

    System.out.println("Enter Attorney's phone number: ex - 4051234567");
    int a_phone = Integer.parseInt(sc.nextLine());
    System.out.println("Enter date client was first assigned: 010112 = Jan 1, 2012");
    int date = Integer.parseInt(sc.nextLine());
    try (final PreparedStatement statement = conn.prepareStatement(Query2))
  {
    // Populate the query template with the data collected from the user
    statement.setInt(1, SSN);
    statement.setString(2, d_name);
    statement.setInt(3, d_phone);
    statement.setString(4, a_name);
    statement.setInt(5, a_phone);
    statement.setInt(6, date);

    final int rows_inserted = statement.executeUpdate();
    System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
  }
    catch(SQLException e)
  {
                  e.printStackTrace();
  }
    displayClient();
    /* Team Association & Active */
    System.out.println("Enter Team Name Associated With:");
    String team = sc.nextLine();
    System.out.println("Client active or inactive?");
    String active = sc.nextLine();

    try (final PreparedStatement statement = conn.prepareStatement(Query2_care))
  {
    // Populate the query template with the data collected from the user
    statement.setInt(1, SSN);
    statement.setString(2, team);
    statement.setString(3, active);

    final int rows_inserted = statement.executeUpdate();
    System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
  }
    catch(SQLException e)
  {
                  e.printStackTrace();
}
    displayCares();
    /* Insurance */
    System.out.println("Enter Insurance Info");
    System.out.println("Enter Policy ID:");
    int pol_id = Integer.parseInt(sc.nextLine());
    System.out.println("Enter Provider ID:");
    int pro_id = Integer.parseInt(sc.nextLine());
    System.out.println("Enter Provider Address:");
    String address = sc.nextLine();
    String type = "";
    boolean constraint_met = false;
    while(!constraint_met)
    {
            System.out.println("Type: \n1.Auto\n2.Life\n3.Health\n4.Home");
    String typed = sc.nextLine();
                    switch(typed) {
                    case "1":
                            type = "Auto";
                            constraint_met = true;
                            break;
                    case "2":
                            type = "Life";
                            constraint_met = true;
                            break;
                    case "3":
                            type = "Health";
                            constraint_met = true;
                            break;
                    case "4":
                            type = "Home";
                            constraint_met = true;
                            break;
                    default:
                            System.out.println("Choose 1, 2, 3, or 4 ONLY");
                            break;
                    }
            }
    // Insert into insurance
    try (final PreparedStatement statement = conn.prepareStatement(Query2_insurance))
```

```
  {
   // Populate the query template with the data collected from the user
   statement.setInt(1, SSN);
   statement.setInt(2, pol_id);
   statement.setInt(3, pro_id);
   statement.setString(4, address);
   statement.setString(5, type);

   final int rows_inserted = statement.executeUpdate();
   System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
  }
   catch(SQLException e)
  {
         e.printStackTrace();
  }

   displayInsurance();
   boolean need_stuff = false;

   int importance_value = 0;
   // Need choices option for multiple needs and a quit option
   while (!need_stuff)
   {
      String need = "";
      System.out.println("Select
Need\n1.Visiting\n2.Shopping\n3.HouseKeeping\n4.Transportation\n5.Yard
Work\n6.Food\n7.Enter yourself\n8.No more needs");
      int select =Integer.parseInt(sc.nextLine());
      switch (select)
      {
      case 1:
      need = "Visiting";
      System.out.println("Enter importance value");
      importance_value = Integer.parseInt(sc.nextLine());
      try (final PreparedStatement statement = conn.prepareStatement(Query2_need))
   {
    // Populate the query template with the data collected from the user
    statement.setInt(1, SSN);
    statement.setString(2, need);
    statement.setInt(3, importance_value);
  int rows_inserted = statement.executeUpdate();
  System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
}
      break;

      case 2:
      need = "Shopping";
      System.out.println("Enter importance value");
      importance_value = Integer.parseInt(sc.nextLine());
      try (final PreparedStatement statement = conn.prepareStatement(Query2_need))
  {
    // Populate the query template with the data collected from the user
    statement.setInt(1, SSN);
    statement.setString(2, need);
    statement.setInt(3, importance_value);

    int rows_inserted = statement.executeUpdate();
    System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
  }
      break;

      case 3:
      need = "HouseKeeping";
      System.out.println("Enter importance value");
      importance_value = Integer.parseInt(sc.nextLine());
      try (final PreparedStatement statement = conn.prepareStatement(Query2_need))
  {
    // Populate the query template with the data collected from the user
    statement.setInt(1, SSN);
    statement.setString(2, need);
    statement.setInt(3, importance_value);

    int rows_inserted = statement.executeUpdate();
    System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
  }
      break;
      case 4:
      need = "Transportation";
      System.out.println("Enter importance value");
```

```
        importance_value =Integer.parseInt(sc.nextLine());
        try (final PreparedStatement statement = conn.prepareStatement(Query2_need))
{
    // Populate the query template with the data collected from the user
    statement.setInt(1, SSN);
    statement.setString(2, need);
    statement.setInt(3, importance_value);

    int rows_inserted = statement.executeUpdate();
    System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
}
        break;
        case 5:
        need = "Yard Work";
        System.out.println("Enter importance value");
        importance_value =Integer.parseInt(sc.nextLine());
        try (final PreparedStatement statement = conn.prepareStatement(Query2_need))
{
    // Populate the query template with the data collected from the user
    statement.setInt(1, SSN);
    statement.setString(2, need);
    statement.setInt(3, importance_value);

    int rows_inserted = statement.executeUpdate();
    System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
}
        break;
        case 6:
        need = "Food";
        System.out.println("Enter importance value");
        importance_value = Integer.parseInt(sc.nextLine());
        try (final PreparedStatement statement = conn.prepareStatement(Query2_need))
{
    // Populate the query template with the data collected from the user
    statement.setInt(1, SSN);
    statement.setString(2, need);
    statement.setInt(3, importance_value);

    int rows_inserted = statement.executeUpdate();
    System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
```

```
        }
            break;
            case 7:
                  System.out.println("What do you need?");
            need = sc.nextLine();
            System.out.println("Enter importance value");
            importance_value = Integer.parseInt(sc.nextLine());
            try (final PreparedStatement statement = conn.prepareStatement(Query2_need))
        {
          // Populate the query template with the data collected from the user
          statement.setInt(1, SSN);
          statement.setString(2, need);
          statement.setInt(3, importance_value);

          int rows_inserted = statement.executeUpdate();
          System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
        }
            break;

            case 8:
            need_stuff = true;
            break;


            }
      }
      // display Need table for project report purpose
      displayNeeds();
}
  public static void Q3() throws SQLException
  {
      // Volunteer
      System.out.println("1. Enter Volunteer\n2. Add Leader");
      int ad = Integer.parseInt(sc.nextLine());
      switch (ad)
      {
      case 1:
      System.out.println("Enter Volunteer SSN:");
      int ssn = Integer.parseInt(sc.nextLine());
      System.out.println("Enter date first join: 010112 - Jan 1, 2012");
```

```
    int date_join = Integer.parseInt(sc.nextLine());
    System.out.println("Enter Recent Training Date");
    int date_train = Integer.parseInt(sc.nextLine());
    System.out.println("Enter Recent Training Location");
    String location = sc.nextLine();
    try (final PreparedStatement statement = conn.prepareStatement(Query3))
  {
    // Populate the query template with the data collected from the user
    statement.setInt(1, ssn);
    statement.setInt(2, date_join);
    statement.setInt(3, date_train);
    statement.setString(4, location);

    int rows_inserted = statement.executeUpdate();
    System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
  }
    // display for pdf purpose
    displayVolunteer();
    // Serve table (relation between team and volunteer)
    System.out.println("Enter Team Associated With:");
    String t_name = sc.nextLine();
    boolean correct = false;
    int month = 0;
    while (!correct)
    {
        System.out.println("Enter month worked");
        month = sc.nextInt();
        if (month > 0 && month < 13)
        {
            correct = true;
        }
        else
        {
            System.out.println("Enter a correct month: 4 - April");
        }
    }

    System.out.println("Enter hours worked ");
    int hour = sc.nextInt();
    sc.nextLine();
```

```
        System.out.println("Active or inactive?");
        String active = sc.nextLine();
        try (final PreparedStatement statement = conn.prepareStatement(Query34_serve))
      {
        // Populate the query template with the data collected from the user
        statement.setInt(1, ssn);
        statement.setString(2, t_name);
        statement.setInt(3, month);
        statement.setInt(4, hour);
        statement.setString(5, active);

        int rows_inserted = statement.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
      }
        // display volunteer/team relation table
        displayServe();
        case 2:
                // add a leader to a team (add to table Leads)
                System.out.println("Enter SSN:");
                int ssn2 = Integer.parseInt(sc.nextLine());
                System.out.println("Team name:");
                String name2 = sc.nextLine();
                try (final PreparedStatement statement = conn.prepareStatement("INSERT INTO
Leads VALUES (" + ssn2 + ",'" + name2 + "')"))
        {
          // Populate the query template with the data collected from the user

          int rows_inserted = statement.executeUpdate();
          System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
        }
                Statement statement2 = conn.createStatement();
                ResultSet r2 = statement2.executeQuery("SELECT * FROM  Leads");
                System.out.println("Leads Table");
                while(r2.next()){

                System.out.println(String.format("SSN: %s| d_name: %s| ",
                r2.getString(1),
                r2.getString(2)));
                        }
        }
```

```
}

public static void Q4() throws SQLException
{
          System.out.println("Enter Volunteer SSN:");
          int ssn2 = sc.nextInt();
          sc.nextLine();
          System.out.println("Enter Team Associated With:");
    String t_name2 = sc.nextLine();
    boolean correct2 = false;
    int month2 = 0;
    while (!correct2)
    {
          System.out.println("Enter month worked");
          month2 = sc.nextInt();
          if (month2 > 0 && month2 < 13)
          {
                correct2 = true;
          }
          else
          {
                System.out.println("Enter a correct month: 4 - April");
          }
    }

    System.out.println("Enter hours worked ");
    int hour2 = sc.nextInt();
    sc.nextLine();
    System.out.println("Active or inactive?");
    String active2 = sc.nextLine();
    try (final PreparedStatement statement = conn.prepareStatement(Query34_serve))
  {
    // Populate the query template with the data collected from the user
    statement.setInt(1, ssn2);
    statement.setString(2, t_name2);
    statement.setInt(3, month2);
    statement.setInt(4, hour2);
    statement.setString(5, active2);

    int rows_inserted = statement.executeUpdate();
    System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
  }
    displayServe();
}

public static void Q5() throws SQLException
{
    System.out.println("1. Enter Employee\n2. Associated with more Team");
    int chose = sc.nextInt();
    sc.nextLine();
    switch(chose)
    {
    // Employee table
    case 1:
    System.out.println("Enter Employee SSN:");
    int ssn = Integer.parseInt(sc.nextLine());
    System.out.println("Enter salary:");
    int salary = Integer.parseInt(sc.nextLine());
    System.out.println("Enter martial status");
    String status = sc.nextLine();
    System.out.println("Enter hired date: 010112 - Jan 1, 2012");
    int date = Integer.parseInt(sc.nextLine());
    try (final PreparedStatement statement = conn.prepareStatement(Query5))
  {
    // Populate the query template with the data collected from the user
    statement.setInt(1, ssn);
    statement.setInt(2, salary);
    statement.setString(3, status);
    statement.setInt(4, date);

    int rows_inserted = statement.executeUpdate();
    System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
  }
    displayEmployee();
    /* // Team associate
    System.out.println("Enter team associated with:");
    String team = sc.nextLine();
    System.out.println("Enter report date: 010112 - Jan 1,2012");
    int report_date = Integer.parseInt(sc.nextLine());
    System.out.println("Enter report description:");
    String des = sc.nextLine();
    try (final PreparedStatement statement = conn.prepareStatement(Query5_report))
  {
    // Populate the query template with the data collected from the user
    statement.setInt(1, ssn);
    statement.setString(2, team);
    statement.setInt(3, report_date);
    statement.setString(4, des);

    int rows_inserted = statement.executeUpdate();
    System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
  }
    displayReport();*/
    break;
    case 2:
          // reporting table for employee and team
          System.out.println("Enter Employee SSN:");
          int ssn2 = Integer.parseInt(sc.nextLine());
          System.out.println("Enter team associated with");
    String team2 = sc.nextLine();
    System.out.println("Enter report description");
    String des2 = sc.nextLine();
    System.out.println("Enter report date ");
    int report_date2 = Integer.parseInt(sc.nextLine());
    //execute query
    try (final PreparedStatement statement = conn.prepareStatement(Query5_report))
    {
      // Populate the query template with the data collected from the user
      statement.setInt(1, ssn2);
      statement.setString(2, team2);
      statement.setInt(3, report_date2);
      statement.setString(4, des2);

      int rows_inserted = statement.executeUpdate();
      System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
    }
    displayReport();
          break;
    }
}
```

```
    public static void Q6() throws SQLException
    {
        // Expenses table
        System.out.println("Enter Employee SSN:");
        int ssn = sc.nextInt();
        sc.nextLine();
        System.out.println("Enter date:");
        int date = sc.nextInt();
        sc.nextLine();
        System.out.println("Enter amount:");
        int amount = sc.nextInt();
        sc.nextLine();
        System.out.println("Enter description:");
        String des = sc.nextLine();
        try (final PreparedStatement statement = conn.prepareStatement(Query6))
      {
        // Populate the query template with the data collected from the user
        statement.setInt(1, ssn);
        statement.setInt(2, date);
        statement.setInt(3, amount);
        statement.setString(4, des);

        int rows_inserted = statement.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
      }
        displayExpense();
    }

    public static void Q7() throws SQLException
    {
        System.out.println("1. Enter a new organization\n2. Associate with team\n3. Affiliate
  with a person");
        int choice = Integer.parseInt(sc.nextLine());
        switch(choice)
        {
        case 1:
            // External Org table
            System.out.println("Enter Orgazniation's Name:");
            String name = sc.nextLine();
            System.out.println("Enter Address:");
            String address = sc.nextLine();
            System.out.println("Enter phone number:");
            int phone = Integer.parseInt(sc.nextLine());
            System.out.println("Enter contact People");
            String contact = sc.nextLine();
            try (final PreparedStatement statement = conn.prepareStatement(Query7))
      {
        // Populate the query template with the data collected from the user
        statement.setString(1, name);
        statement.setString(2, address);
        statement.setInt(3, phone);
        statement.setString(4, contact);

        int rows_inserted = statement.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
      }
            // display External Org table after insert
            displayOrg();
            break;
      case 2:
            // Sponsor table
            System.out.println("Enter Orgazniation Name:");
            String name2 = sc.nextLine();
            System.out.println("Enter Team Name:");
            String t_name = sc.nextLine();
            try (final PreparedStatement statement =
conn.prepareStatement(Query7_sponsor))
      {
        // Populate the query template with the data collected from the user
        statement.setString(1, name2);
        statement.setString(2, t_name);
        int rows_inserted = statement.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
      }
            // display sponsor table after insert
            displaySponsor();
            break;
      case 3:
            break;
      }
  }
  public static void Q8() throws SQLException
  {
      //displayDCard();
      System.out.println("1. Enter a new donor\n2. Associate donor with 1 or more
Donation(s)");
      int choice = Integer.parseInt(sc.nextLine());
      switch (choice)
      {
      case 1:
            // Donor table
            System.out.println("Enter SSN:");
            int ssn = Integer.parseInt(sc.nextLine());
            System.out.println("Is it anonymous");
            String ano = sc.nextLine();
            try (final PreparedStatement statement = conn.prepareStatement(Query8))
      {
        // Populate the query template with the data collected from the user
        statement.setInt(1, ssn);
        statement.setString(2, ano);
        int rows_inserted = statement.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
      }
            displayDonor();
            break;
      case 2:
            // info for donor card or check
            System.out.println("Donor SSN:");
            int ssn2 = Integer.parseInt(sc.nextLine());
            System.out.println("Date:");
            int date = Integer.parseInt(sc.nextLine());
            System.out.println("Amount");
            int amount = Integer.parseInt(sc.nextLine());
            System.out.println("Campaign:");
            String camp = sc.nextLine();
            System.out.println("Type:\n1. Card\n2. Check");
            int type = Integer.parseInt(sc.nextLine());
            boolean chose = false;
        while (!chose)
        {
            switch (type)
            {
            case 1:
                // donor card table
                String c = "card";
                System.out.println("Card number:");
                int card = Integer.parseInt(sc.nextLine());
                System.out.println("Card type:");
                String c_type = sc.nextLine();
                System.out.println("Exp date: 010112 - Jan 1, 2012");
                int exp = Integer.parseInt(sc.nextLine());
                // add to the donor donate table
                try (final PreparedStatement statement =
conn.prepareStatement(Query8_Donate))
        {
          // Populate the query template with the data collected from the user
          statement.setInt(1, ssn2);
          statement.setInt(2, date);
          statement.setInt(3, amount);
          statement.setString(4, c);
          statement.setString(5, camp);
          int rows_inserted = statement.executeUpdate();
          System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
        }
                // add to donor donate card table
                try (final PreparedStatement statement =
conn.prepareStatement(Query8_Card))
        {
          // Populate the query template with the data collected from the user
          statement.setInt(1, ssn2);
          statement.setInt(2, date);
          statement.setInt(3, amount);
          statement.setInt(4, card);
          statement.setString(5, c_type);
          statement.setInt(6, exp);
          int rows_inserted = statement.executeUpdate();
          System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
        }
```

```
                        chose = true;

                        displayDCard();
                        break;
                    case 2:
                        // add to donor check info
                        String ch = "check";
                        System.out.println("Check number:");
                        int check = Integer.parseInt(sc.nextLine());
                        // add to table donor donate
                        try (final PreparedStatement statement =
conn.prepareStatement(Query8_Donate))
                {
                    // Populate the query template with the data collected from the user
                    statement.setInt(1, ssn2);
                    statement.setInt(2, date);
                    statement.setInt(3, amount);
                    statement.setString(4, ch);
                    statement.setString(5, camp);
                    int rows_inserted = statement.executeUpdate();
                    System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
                }
                        displayDD();
                        // add to table donor donate check
                        try (final PreparedStatement statement =
conn.prepareStatement(Query8_Check))
                {
                    // Populate the query template with the data collected from the user
                    statement.setInt(1, ssn2);
                    statement.setInt(2, date);
                    statement.setInt(3, amount);
                    statement.setInt(4, check);
                    int rows_inserted = statement.executeUpdate();
                    System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
                }
                        chose = true;

                        displayDCH();
                        break;
                    default:
                        System.out.println("Chose 1 or 2");
                        break;
                  }

            }

            break;
      }
  }
  // basically mirroring Q8 except adding into different tables
  public static void Q9() throws SQLException
  {
      System.out.println("1. Enter new org that donates \n2. Associated with 1 or more
Donation(s)");
      int choice = Integer.parseInt(sc.nextLine());
      switch (choice)
      {
      case 1:
            // Org Donate table
            System.out.println("Enter Org name:");
            String name =sc.nextLine();
            System.out.println("Is it anonymous");
            String ano = sc.nextLine();
            try (final PreparedStatement statement = conn.prepareStatement(Query9))
      {
        // Populate the query template with the data collected from the user
        statement.setString(1, name);
        statement.setString(2, ano);
        int rows_inserted = statement.executeUpdate();
        System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
      }
            displayOD();
            break;
      case 2:
            System.out.println("Org Name:");
            String o_name = sc.nextLine();
            System.out.println("Date:");
            int d = Integer.parseInt(sc.nextLine());
            System.out.println("Amount");
            int amount = Integer.parseInt(sc.nextLine());
        System.out.println("Campaign:");
        String camp = sc.nextLine();
        System.out.println("Type:\n1. Card\n2. Check");
        int type = Integer.parseInt(sc.nextLine());
        boolean chose = false;
        while (!chose)
        {
            switch (type)
            {
            case 1:
                String c = "card";
                System.out.println("Card number:");
                int card = Integer.parseInt(sc.nextLine());
                System.out.println("Card type:");
                String c_type = sc.nextLine();
                System.out.println("Exp date: 010112 - Jan 1, 2012");
                int exp = Integer.parseInt(sc.nextLine());
                try (final PreparedStatement statement =
conn.prepareStatement(Query9_donate))
        {
          // Populate the query template with the data collected from the user
          statement.setString(1, o_name);
          statement.setInt(2, d);
          statement.setInt(3, amount);
          statement.setString(4, c);
          statement.setString(5, camp);
          int rows_inserted = statement.executeUpdate();
          System.out.println(String.format("Done. %d rows inserted.", rows_inserted));

        }
                displayOD();
                try (final PreparedStatement statement =
conn.prepareStatement(Query9_Card))
        {
          // Populate the query template with the data collected from the user
          statement.setString(1, o_name);
          statement.setInt(2, d);
          statement.setInt(3, amount);
          statement.setInt(4, card);
          statement.setString(5, c_type);
```

```
                   statement.setInt(6, exp);
                   int rows_inserted = statement.executeUpdate();
                   System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
                }
                                chose = true;

                                displayODCard();
                                break;
                        case 2:
                                String ch = "check";
                                System.out.println("Check number:");
                                int check = Integer.parseInt(sc.nextLine());
                                try (final PreparedStatement statement =
conn.prepareStatement(Query9_donate))
                {
                   // Populate the query template with the data collected from the user
                   statement.setString(1, o_name);
                   statement.setInt(2, d);
                   statement.setInt(3, amount);
                   statement.setString(4, ch);
                   statement.setString(5, camp);
                   int rows_inserted = statement.executeUpdate();
                   System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
                }
                                displayOD();
                                try (final PreparedStatement statement =
conn.prepareStatement(Query9_Check))
                {
                   // Populate the query template with the data collected from the user
                   statement.setString(1, o_name);
                   statement.setInt(2, d);
                   statement.setInt(3, amount);
                   statement.setInt(4, check);
                   int rows_inserted = statement.executeUpdate();
                   System.out.println(String.format("Done. %d rows inserted.", rows_inserted));
                }
                                chose = true;

                                displayODCheck();
                                break;
```

```
                    default:
                            System.out.println("Chose 1 or 2");
                            break;
                    }

            }

            break;
        }

    }
    public static void Q10() throws SQLException
    {

        System.out.println("Enter SSN of Client:");
        int SSN = sc.nextInt();
        String q10 = "SELECT SSN,d_name,d_phone FROM Client WHERE SSN ="+SSN;
        Statement statement = conn.createStatement();
                ResultSet r = statement.executeQuery(q10);

                        //display
                while(r.next()){
                        System.out.println(String.format("SSN: %s|Doctor name: %s | Phone %s
|",
              r.getString("SSN"),
              r.getString("d_name"),
              r.getString("d_phone")));
                }
    }
    public static void Q11() throws SQLException
    {
        System.out.println("Expense date:");
        int date = Integer.parseInt(sc.nextLine());
        String q10 = "SELECT SSN, amount FROM Expenses WHERE e_date ="+date;
        Statement statement = conn.createStatement();
                ResultSet r = statement.executeQuery(q10);

                        //display
                while(r.next()){
                        System.out.println(String.format("SSN: %s|Amount: %s |",
              r.getString("SSN"),
              r.getString("amount")));
                }
    }
    public static void Q12() throws SQLException
    {
        System.out.println("Enter client's SSN");
        int ssn = Integer.parseInt(sc.nextLine());
        String q12 = "SELECT DISTINCT (SSN) FROM Serves WHERE t_name in(SELECT
t_name from Cares WHERE SSN in(SELECT SSN FROM Client where SSN="+ ssn +"))";
        Statement statement = conn.createStatement();
                ResultSet r = statement.executeQuery(q12);

        // display
                while(r.next()){
        System.out.println(String.format("SSN: %s|",
        r.getString("SSN")));
                }
    }
    public static void Q13() throws SQLException
    {
        String q13 = "Select P_name, M_address, E_address, H_number, W_number, C_number
FROM People where SSN IN(SELECT SSN from Cares where t_name " +
                  "IN(SELECT t_name FROM Sponsor WHERE org_name IN(SELECT
org_name FROM External_Organization WHERE SUBSTRING(org_name,1,1) between 'B' and
'K'))) " +
                  "ORDER BY P_name";
        Statement statement = conn.createStatement();
                ResultSet r = statement.executeQuery(q13);
        while(r.next()){

                System.out.println(String.format("name: %s mail: %s email: %s home: %s work:
%s cell: %s",
                r.getString("P_name"),
                r.getString("M_address"),
                r.getString("E_address"),
                r.getString("H_number"),
                r.getString("W_number"),
                r.getString("C_number")));
                  }
  }
  public static void Q14() throws SQLException
  {
        String q13 = "SELECT People.P_name, SUM(Donor_Donate.d_amount) AS
total_amount, Donor.anonymous FROM ((Donor INNER JOIN Donor_Donate ON Donor.SSN =
Donor_Donate.SSN)" +
                              " INNER JOIN People ON Donor.SSN = People.SSN) WHERE
Donor.SSN IN(SELECT SSN FROM Employee) GROUP BY Donor.anonymous,
People.P_name Order by total_amount";
        Statement statement = conn.createStatement();
              ResultSet r = statement.executeQuery(q13);
        while(r.next()){

              System.out.println(String.format("name: %s| total amount: %s| anonymous: %s",
              r.getString("P_name"),
              r.getString("total_amount"),
              r.getString("anonymous")));
                    }
  }
  public static void Q15() throws SQLException
  {
        System.out.println("Enter date:");
        String date = sc.nextLine();//sc.nextLine();
        // get first 2 number for month
        char m1 = date.charAt(0);
        char m2 = date.charAt(1);
        // combine string
        String month = String.valueOf(m1) + String.valueOf(m2);
        //back to int for comparison
        int m = Integer.parseInt(month);
        // repeat for day
        char d1 = date.charAt(2);
        char d2 = date.charAt(3);
        String day = String.valueOf(d1) + String.valueOf(d2);
        int d = Integer.parseInt(day);
        // repeat for year
        char y1 = date.charAt(4);
        char y2 = date.charAt(5);
        String year = String.valueOf(y1) + String.valueOf(y2);
        int y = Integer.parseInt(year);
        String q15 = "SELECT * FROM Team";
        Statement statement = conn.createStatement();
                ResultSet r = statement.executeQuery(q15);
        while(r.next()){
                String date_formed = r.getString("date_formed");
                //System.out.println(date_formed);
                char m12 = date_formed.charAt(0);
        char m22 = date_formed.charAt(1);
        String month2 = String.valueOf(m12) + String.valueOf(m22);
        int mon = Integer.parseInt(month2);

        char d12 = date_formed.charAt(2);
        char d22 = date_formed.charAt(3);
        String day2 = String.valueOf(d12) + String.valueOf(d22);
        int da = Integer.parseInt(day2);

        char y12 = date_formed.charAt(4);
        char y22 = date_formed.charAt(5);
        String year2 = String.valueOf(y12) + String.valueOf(y22);
        int ye = Integer.parseInt(year2);
        // compare input date and date in table
        if (ye > y || (ye > y && mon > m) || (ye > y && mon > m && da > d))
        {
                System.out.println(String.format("name: %s|",
                r.getString("t_name")));
                        }
        }

   }
   public static void Q16() throws SQLException
   {
        String q16 = "UPDATE Employee SET salary=salary*1.10 WHERE SSN IN(SELECT
SSN FROM Reporting GROUP BY SSN HAVING count(t_name)>1)";
        Statement statement = conn.createStatement();
        statement.executeUpdate(q16);
                ResultSet r = statement.executeQuery("SELECT * FROM  Employee");
        while(r.next()){

                System.out.println(String.format("SSN: %s| Salary: %s|",
                r.getString("SSN"),
      r.getString("salary")));
         }
 }
 public static void Q17() throws SQLException
 {
   String q17 = "Delete FROM Client WHERE (Client.SSN IN(SELECT SSN FROM
Needs WHERE Client.SSN = Needs.SSN AND Needs.needs = 'Transportation' AND
Needs.importance_value < 5)) " +
         "AND (Client.SSN NOT IN(SELECT SSN FROM Insurance_Policy
WHERE i_type = 'Health'))";
   Statement statement = conn.createStatement();
   statement.executeUpdate(q17);
   ResultSet r = statement.executeQuery("SELECT * FROM  Client");
   while(r.next()){

      System.out.println(String.format("SSN: %s| d_name: %s| d_phone: %s| a_name:
%s| a_name: %s| type: %s|",
      r.getString(1),
      r.getString(2),
      r.getString(3),
      r.getString(4),
      r.getString(5),
      r.getString(5)));
         }
 }

 public static void Q18() throws SQLException
 {
   Statement statement = conn.createStatement();
   String name;
   String type;
    String date;
    try {
      BufferedReader bufferedReader = new BufferedReader(new FileReader(new
File("file.txt")));
     String line = null;
     // read till there's nothing to read
      while ((line = bufferedReader.readLine()) != null) {
            String tmp[]=line.split(",");
             name=tmp[0];
```

```
                                   type=tmp[1];
                                   date=tmp[2];
                                         String q18 = "INSERT INTO Team
(t_name,t_type,date_formed) values ('" + name + "','" + type + "',"+ date + ")";

                                         statement.executeUpdate(q18);

                  }
                  bufferedReader.close();
                 } catch (IOException e) {
                  e.printStackTrace();
                 }
           // display table with data add in
           ResultSet r = statement.executeQuery("SELECT * FROM  Team");
        while(r.next()){

                System.out.println(String.format("Name: %s| type: %s| date: %s| ",
                r.getString(1),
                r.getString(2),
                r.getString(3)));
                        }

   }

   public static void Q19() throws SQLException
   {
        Statement statement = conn.createStatement();
        String file_export = "export.txt";
                File file = new File(file_export);
                FileWriter fw = null;
                try {
                        fw = new FileWriter(file.getAbsoluteFile());
                        BufferedWriter bw = new BufferedWriter(fw);
                        // All the information you need from someone XD
                        String sql = "SELECT SSN, P_name, E_address, C_number FROM
People WHERE on_list='yes'";
                        try {

                                ResultSet r = statement.executeQuery(sql);
                                while(r.next()) {
                                        bw.write(r.getString(1) + "," +r.getString(2) + ","
+r.getString(3) + "," +r.getString(4) +"\n");
                                        System.out.println("Write in file export");
                                }

                        }catch(IOException | SQLException e ) {
                                //bw.close();
                                e.printStackTrace();
                        }finally {
                                try {bw.close();} catch (Exception ex) {}
                         }
                }catch(IOException e) {
                        e.printStackTrace();
                }
  }

  public static void Q20() throws SQLException
  {
        Statement statement = conn.createStatement();
        System.out.println("1. Enter Person\n2. Affiliate with an Org");
        int choice = Integer.parseInt(sc.nextLine());
        switch (choice)
        {
        case 1:
        System.out.println("Enter SSN");
        int ssn = Integer.parseInt(sc.nextLine());
        System.out.println("Enter name");
        String name = sc.nextLine();
        System.out.println("Enter date of birth: 010101-Jan 1,2001");
        int dob = Integer.parseInt(sc.nextLine());
        System.out.println("Enter race");
        String race = sc.nextLine();
        System.out.println("Enter gender");
        String gender = sc.nextLine();
        System.out.println("Enter profession");
        String prof = sc.nextLine();
        System.out.println("Enter mail address");
        String m_a = sc.nextLine();
        System.out.println("Enter email address");
        String e_a = sc.nextLine();
```

```
    System.out.println("Enter home phone number");
    String h_num = sc.nextLine();
    System.out.println("Enter work phone number");
    String w_num = sc.nextLine();
    System.out.println("Enter Cell phone number");
    String c_num = sc.nextLine();
    System.out.println("On mailling list?");
    String onlist = sc.nextLine();

    String q12 = "INSERT INTO People VALUES
("+ssn+",'"+name+"',"+dob+",'"+race+"','"+gender+"','"+prof+"','"+m_a+"','"+e_a+"',"
    + h_num+ ","+w_num+","+c_num+",'"+onlist+"')";

        int re = statement.executeUpdate(q12);
        System.out.println(String.format("Done. %d rows inserted.", re));
    ResultSet r = statement.executeQuery("SELECT * FROM  People");
    while(r.next()){

        System.out.println(String.format("SSN: %s| name: %s| dob: %s| race: %s| gender:
%s| prof: %s|" +
                "mail: %s| email: %s| home: %s| work: %s| cel: %s|",
        r.getString(1),
        r.getString(2),
        r.getString(3),
        r.getString(4),
        r.getString(5),
        r.getString(6),
        r.getString(7),
        r.getString(8),
        r.getString(9),
        r.getString(10),
        r.getString(11)));
            }
    break;
    case 2:
        System.out.println("Enter SSN:");
        int ssn2 = Integer.parseInt(sc.nextLine());
        System.out.println("Enter Org name");
        String org = sc.nextLine();
```

```
            int re2 = statement.executeUpdate("INSERT INTO Affiliates VALUES (" + ssn2
+ ",'" + org + "')");
            System.out.println(String.format("Done. %d rows inserted.", re2));
      ResultSet r2 = statement.executeQuery("SELECT * FROM  Affiliates");
      while(r2.next()){

            System.out.println(String.format("SSN: %s| org: %s|",
            r2.getString(1),
            r2.getString(2)));
                  }

      }
  }
  public static void displayDCard() throws SQLException
  {
      Statement statement = conn.createStatement();
    ResultSet r = statement.executeQuery("SELECT * from Donor_Donate_Card");

                  //display
    while(r.next())
    {
                  System.out.println("\nContent of Donor Card table");
                  System.out.println("\nSSN | date | amount|card num|card type| exp date|");
                  System.out.println(String.format("%s | %s | %s | %s | %s | %s |",
          r.getString(1),
          r.getString(2),
          r.getString(3),
          r.getString(4),
          r.getString(5),
          r.getString(6)));
    }

  }

  public static void displayDCH()
  {
      String sql = "Select * from Donor_Donate_Check";
      try (final Statement statement = conn.createStatement();
            ResultSet r = statement.executeQuery(sql)) {
                  //display
                  System.out.println("\nContent of Donor Check table");
                  System.out.println("\nSSN | date | amount | check num|");
                  while(r.next()) {
                        System.out.println(String.format("%s | %s |%s | %s |",
            r.getString(1),
            r.getString(2),
            r.getString(3),
            r.getString(4)));
                  }
            }
    catch(SQLException e)
    {
                  e.printStackTrace();
            }
  }


  public static void displayDD()
  {
      String sql = "Select * from Donor_Donate";
      try (final Statement statement = conn.createStatement();
            ResultSet r = statement.executeQuery(sql)) {

                  //display
                  System.out.println("\nContent of Donate table");
                  System.out.println("\nSSN | date | amount|type|camp");
                  while(r.next()) {
                        System.out.println(String.format("%s | %s |%s | %s |%s | ",
            r.getString(1),
            r.getString(2),
            r.getString(3),
            r.getString(4),
            r.getString(5)));
                  }
            }
    catch(SQLException e)
    {
                  e.printStackTrace();
            }
```

```
}

public static void displayDonor()
{
    String sql = "Select * from Donor";
    try (final Statement statement = conn.createStatement();
          ResultSet r = statement.executeQuery(sql)) {

                //display
                System.out.println("\nContent of Donor table");
                System.out.println("\nSSN | anonymous |");
                while(r.next()) {
                      System.out.println(String.format("%s | %s |",
          r.getString(1),
          r.getString(2)));
                }
          }
  catch(SQLException e)
  {
                e.printStackTrace();
          }
}

public static void displaySponsor()
{
    String sql = "Select * from Sponsor";
    /* display team for image report purpose */
  try (final Statement statement = conn.createStatement();
          ResultSet r = statement.executeQuery(sql)) {

                //display
                System.out.println("\nContent of Sponsor table");
                System.out.println("\nOrg | Team |");
                while(r.next()) {
                      System.out.println(String.format("%s | %s |",
          r.getString(1),
          r.getString(2)));
                }
          }
  catch(SQLException e)
  {
                e.printStackTrace();
          }
}

public static void displayOrg()
{
    String sql = "Select * from External_Organization";
    /* display team for image report purpose */
  try (final Statement statement = conn.createStatement();
          ResultSet r = statement.executeQuery(sql)) {

                //display
                System.out.println("\nContent of Org table");
                System.out.println("\nName | address | phone| contact |");
                while(r.next()) {
                      System.out.println(String.format("%s | %s | %s | %s |",
          r.getString(1),
          r.getString(2),
          r.getString(3),
          r.getString(4)));
                }
          }
  catch(SQLException e)
  {
                e.printStackTrace();
          }
}

public static void displayExpense()
{
    String sql = "Select * from Expenses";
    /* display team for image report purpose */
  try (final Statement statement = conn.createStatement();
          ResultSet r = statement.executeQuery(sql)) {

                //display
                System.out.println("\nContent of Expense table");
                System.out.println("\nSSN | date | Amount| description |");
                while(r.next()) {
```

```
                                System.out.println(String.format("%s | %s | %s | %s |",
                r.getString(1),
                r.getString(2),
                r.getString(3),
                r.getString(4)));
                        }
                }
   catch(SQLException e)
   {
                        e.printStackTrace();
                }
}

/* for result purpose */
public static void displayTeam()
{
      String sql = "Select * from Team";
      /* display team for image report purpose */
   try (final Statement statement = conn.createStatement();
                ResultSet r = statement.executeQuery(sql)) {

                        //display
                        System.out.println("\nContent of Team table");
                        System.out.println("\nTeam Name | Team Type | Date Formed");
                        while(r.next()) {
                                System.out.println(String.format("%s | %s | %s | ",
                r.getString(1),
                r.getString(2),
                r.getString(3)));
                        }
                }
   catch(SQLException e)
   {
                        e.printStackTrace();
                }
}

public static void displayClient()
{
      String sql = "SELECT * FROM Client";
      /* display team for image report purpose */
    try (final Statement statement = conn.createStatement();
            ResultSet r = statement.executeQuery(sql)) {

                  //display
                  System.out.println("\nContent of Client table");
                  System.out.println("\nSSN | d_name | d_phone | a_name | a_phone | date
first assigned");
                  while(r.next()) {
                        System.out.println(String.format("%s | %s | %s | %s | %s | %s |",
            r.getString(1),
            r.getString(2),
            r.getString(3),
            r.getString(4),
            r.getString(5),
            r.getString(6)));
                  }
            }
    catch(SQLException e)
    {
                  e.printStackTrace();
            }
  }

  public static void displayCares()
  {
      String sql = "SELECT * FROM Cares";
      /* display team for image report purpose */
    try (final Statement statement = conn.createStatement();
            ResultSet r = statement.executeQuery(sql)) {

                  //display
                  System.out.println("\nContent of Care table");
                  System.out.println("\nSSN | Team Name | Active");
                  while(r.next()) {
                        System.out.println(String.format("%s | %s | %s | ",
            r.getString(1),
            r.getString(2),
            r.getString(3)));
                  }
```

```
          }
  catch(SQLException e)
  {
                e.printStackTrace();
          }
}

public static void displayNeeds()
{
    String sql = "SELECT * FROM Needs";
    /* display team for image report purpose */
  try (final Statement statement = conn.createStatement();
          ResultSet r = statement.executeQuery(sql)) {

                //display
                System.out.println("\nContent of Needs table");
                System.out.println("\nSSN | Needs | Importance Value");
                while(r.next()) {
                      System.out.println(String.format("%s | %s | %s | ",
          r.getString(1),
          r.getString(2),
          r.getString(3)));
                }
          }
  catch(SQLException e)
  {
                e.printStackTrace();
          }
}

public static void displayInsurance()
{
    String sql = "SELECT * FROM Insurance_Policy";
    /* display team for image report purpose */
  try (final Statement statement = conn.createStatement();
          ResultSet r = statement.executeQuery(sql)) {

                //display
                System.out.println("\nContent of Insurance Policy table");
                System.out.println("\nSSN | pol_id | pro_id | pro_ad | i_type |");
                while(r.next()) {
                      System.out.println(String.format("%s | %s | %s | %s | %s |",
          r.getString(1),
          r.getString(2),
          r.getString(3),
          r.getString(4),
          r.getString(5)));
                }
          }
  catch(SQLException e)
  {
                e.printStackTrace();
          }
}
public static void displayVolunteer()
{
    String sql = "SELECT * FROM Volunteers";
    /* display team for image report purpose */
  try (final Statement statement = conn.createStatement();
          ResultSet r = statement.executeQuery(sql)) {

                //display
                System.out.println("\nContent of Volunteer table");
                System.out.println("\nSSN | date_first join | date recent train | location |");
                while(r.next()) {
                      System.out.println(String.format("%s | %s | %s | %s |",
          r.getString(1),
          r.getString(2),
          r.getString(3),
          r.getString(4)));
                }
          }
  catch(SQLException e)
  {
                e.printStackTrace();
          }
}

public static void displayServe()
{
        String sql = "SELECT * FROM Serves";
        /* display team for image report purpose */
    try (final Statement statement = conn.createStatement();
                ResultSet r = statement.executeQuery(sql)) {

                        //display
                        System.out.println("\nContent of Insurance Policy table");
                        System.out.println("\nSSN | name | month | hour | active |");
                        while(r.next()) {
                                System.out.println(String.format("%s | %s | %s | %s | %s |",
                r.getString(1),
                r.getString(2),
                r.getString(3),
                r.getString(4),
                r.getString(5)));
                        }
                }
    catch(SQLException e)
    {
                        e.printStackTrace();
                }
}

public static void displayActive()
{
        String sql = "SELECT * FROM Active";
        /* display team for image report purpose */
    try (final Statement statement = conn.createStatement();
                ResultSet r = statement.executeQuery(sql)) {

                        //display
                        System.out.println("\nContent of Active table");
                        System.out.println("\nSSN | team | active |");
                        while(r.next()) {
                                System.out.println(String.format("%s | %s | %s |",
                r.getString(1),
                r.getString(2),
                r.getString(3)));
                        }
                }
```

```
  catch(SQLException e)
  {
            e.printStackTrace();
        }
}

public static void displayEmployee()
{
    String sql = "SELECT * FROM Employee";
    /* display team for image report purpose */
  try (final Statement statement = conn.createStatement();
        ResultSet r = statement.executeQuery(sql)) {

            //display
            System.out.println("\nContent of Employee table");
            System.out.println("\nSSN | salary | martial status | hire date |");
            while(r.next()) {
                System.out.println(String.format("%s | %s | %s | %s |",
        r.getString(1),
        r.getString(2),
        r.getString(3),
        r.getString(4)));
            }
        }
  catch(SQLException e)
  {
            e.printStackTrace();
        }
}

public static void displayReport()
{
    String sql = "SELECT * FROM Reporting";
    /* display team for image report purpose */
  try (final Statement statement = conn.createStatement();
        ResultSet r = statement.executeQuery(sql)) {

            //display
            System.out.println("\nContent of Reporting table");
            System.out.println("\nSSN | team | date | description |");
```

```
                        while(r.next()) {
                                 System.out.println(String.format("%s | %s | %s | %s |",
               r.getString(1),
               r.getString(2),
               r.getString(3),
               r.getString(4)));
                        }
               }
   catch(SQLException e)
   {
                        e.printStackTrace();
               }
}
public static void displayOD() throws SQLException
{
      Statement statement = conn.createStatement();
   ResultSet r = statement.executeQuery("SELECT * FROM Organization_Donor");

                        //display
   while(r.next())
   {
                        System.out.println("\nContent of Donor Card table");
                        System.out.println("\nOrg Name | anonymous |");
                        System.out.println(String.format("%s | %s |",
            r.getString(1),
            r.getString(2)));
   }
      }
public static void displayODCheck() throws SQLException
{
      Statement statement = conn.createStatement();
   ResultSet r = statement.executeQuery("SELECT * FROM Organization_Donate_Check");

                        //display
   while(r.next())
   {
                        System.out.println("\nContent of Org Card table");
                        System.out.println("\nSSN | date | amount | check num|");
                        System.out.println(String.format("%s | %s |%s | %s |",
            r.getString(1),
          r.getString(2),
          r.getString(3),
          r.getString(4)));
    }
      }
  public static void displayODCard() throws SQLException
  {
      Statement statement = conn.createStatement();
    ResultSet r = statement.executeQuery("SELECT * FROM Organization_Donate_Card");

                  //display
    while(r.next())
    {
                  System.out.println("\nContent of Org Card table");
                  System.out.println("\nOrg Name | date | amount|card num|card type| exp
date|");
                  System.out.println(String.format("%s | %s |%s | %s |%s | %s |",
                              r.getString(1),
          r.getString(2),
          r.getString(3),
          r.getString(4),
          r.getString(5),
          r.getString(6)));
    }
      }
}
```

<u>Task 6:</u> Test

Note: I added another Query option called **20. Add People** because it is relevant to this project since the People table is basically the center of all tables. The **Close/Quit** option is moved to **Query 21**.

Query 1:

- Please read Query 2 description carefully as I have a detailed explanation. I did not put it in Query 1 because it was too simple to run compared to Query 2-17.

```
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
```

```
1
Please enter team name:
Yellow
Please enter team type:
medical
Please enter date formed: 010112 = Jan 1, 2012
010101
Done. 1 rows inserted.

Content of Team table

Team Name | Team Type | Date Formed
Yellow | medical | 010101 |
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for transportation
```

```
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
1
Please enter team name:
Red
Please enter team type:
kitchen
Please enter date formed: 010112 = Jan 1, 2012
010102
Done. 1 rows inserted.

Content of Team table

Team Name | Team Type | Date Formed
Red | kitchen | 010102 |
Yellow | medical | 010101 |
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
```

```
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
1
Please enter team name:
Blue
Please enter team type:
build
Please enter date formed: 010112 = Jan 1, 2012
010203
Done. 1 rows inserted.

Content of Team table

Team Name | Team Type | Date Formed
Blue  | build | 010203 |
Red | kitchen | 010102 |
Yellow | medical | 010101 |
```

Query 2:

Before running:

Add people to the people table with Query 20, but for the sake of not adding 100+ lines of java output for each person added I will just put the INSERT function in SQL so you can see what is already in the People table.
Note that I will use this way to show output for* ***all Query after this (Query 2-17) *that needs it. I will only show the Java output for function relevance. For example, in Query 2, I will not show how I add data to the People table because the output takes up a lot of lines and because that's the output of Query 20. I will also not show how I add Team because that's Query 1. I will instead show it in sql form for simplicity and so you can understand quickly what I added before running Query 2. Then I will show output of how I added data to the Clients, Needs and Policy because that's what my Query 2 code does.*
***Note that I will only show the sql table that are relevant. For example, while most tables are filled, Query 6 only requires People and Employee table to run, therefore, I will only show those 2 tables in insert sql form.*

```
INSERT INTO People VALUES
(1,'john',010100, 'white', 'm', 'dance', '11 robin rd', 'john@mail.com', 123456789, 234567890,
345678900, 'no')
INSERT INTO People VALUES
(2,'po',010110, 'asian', 'm', 'cs', '12 robinson rd', 'po@mail.com', 123456789, 234567890,
345678900, 'yes')
INSERT INTO People VALUES
(3,'cait',010205, 'white', 'f', 'ee', '1 king rd', 'cait@mail.com', 123445789, 2385676,
345678900, 'no')
INSERT INTO People VALUES
(4,'grace',010313, 'asian', 'f', 'music', '12 low rd', 'grace@mail.com', 123456789, 4582745,
6476583, 'yes')
INSERT INTO People VALUES
(5,'brand',010303, 'black', 'm', 'ame', '1 lowes rd', 'brand@mail.com', 123346789, 4672745,
6487583, 'yes'),
(6, 'sam', 010303, 'native', 'f', 'med', '12 lowes rd', 'sam@mail.com', 543346789, 4634745,
6457583, 'yes')

 INSERT INTO Team VALUES
('Yellow', 'medical', '010101'),
('Red', 'kitchen', '010102'),
('Blue', 'build', '010203')
```

Result:
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team

```
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
2
Enter client SSN from People table:
1
Enter Doctor's name
ben
Enter Doctor's phone number: ex - 4051234567
1234567
Enter Attorney's name
son
Enter Attorney's phone number: ex - 4051234567
3456789
Enter date client was first assigned: 010112 = Jan 1, 2012
010104
```

```
Done. 1 rows inserted.

Content of Client table

SSN | d_name | d_phone | a_name | a_phone | date first assigned
1 | ben | 1234567 | son | 3456789 | 10104 |
Enter Team Name Associated With:
Yellow
Client active or inactive?
active
Done. 1 rows inserted.

Content of Care table

SSN | Team Name | Active
1 | Yellow | active |
Enter Insurance Info
Enter Policy ID:
34
Enter Provider ID:
43
Enter Provider Address:
ad1
Type:
1.Auto
2.Life
3.Health
4.Home
Auto
Choose 1, 2, 3, or 4 ONLY
Type:
1.Auto
2.Life
3.Health
4.Home
1
Done. 1 rows inserted.

Content of Insurance Policy table
```

```
SSN | pol_id | pro_id | pro_ad | i_type |
1 | 34 | 43 | ad1 | Auto |
Select Need
1.Visiting
2.Shopping
3.HouseKeeping
4.Transportation
5.Yard Work
6.Food
7.Enter yourself
8.No more needs
4
Enter importance value
6
Done. 1 rows inserted.
Select Need
1.Visiting
2.Shopping
3.HouseKeeping
4.Transportation
5.Yard Work
6.Food
7.Enter yourself
8.No more needs
1
Enter importance value
6
Done. 1 rows inserted.
Select Need
1.Visiting
2.Shopping
3.HouseKeeping
4.Transportation
5.Yard Work
6.Food
7.Enter yourself
8.No more needs
8
```

Content of Needs table

```
SSN | Needs | Importance Value
1 | Transportation | 6 |
1 | Visiting | 6 |
```

Query 3:

Before running:

```
INSERT INTO People VALUES
```

```
(1,'john',010100, 'white', 'm', 'dance', '11 robin rd', 'john@mail.com', 123456789, 234567890,
345678900, 'no')
INSERT INTO People VALUES
(2,'po',010110, 'asian', 'm', 'cs', '12 robinson rd', 'po@mail.com', 123456789, 234567890,
345678900, 'yes')
INSERT INTO People VALUES
(3,'cait',010205, 'white', 'f', 'ee', '1 king rd', 'cait@mail.com', 123445789, 2385676,
345678900, 'no')
INSERT INTO People VALUES
(4,'grace',010313, 'asian', 'f', 'music', '12 low rd', 'grace@mail.com', 123456789, 4582745,
6476583, 'yes')
INSERT INTO People VALUES
(5,'brand',010303, 'black', 'm', 'ame', '1 lowes rd', 'brand@mail.com', 123346789, 4672745,
6487583, 'yes'),
(6, 'sam', 010303, 'native', 'f', 'med', '12 lowes rd', 'sam@mail.com', 543346789, 4634745,
6457583, 'yes')

 INSERT INTO Team VALUES
('Yellow', 'medical', '010101'),
('Red', 'kitchen', '010102'),
('Blue', 'build', '010203')
```

Result:

Enter you option(1-17):

1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams sponsored by

```
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
3
1. Enter Volunteer
2. Add Leader
1
Enter Volunteer SSN:
3
Enter date first join: 010112 - Jan 1, 2012
010113
Enter Recent Training Date
010119
Enter Recent Training Location
NV
Done. 1 rows inserted.

Content of Volunteer table

SSN | date_first join | date recent train | location |
3 | 10113 | 10119 | NV |
Enter Team Associated With:
Red
```

```
Enter month worked
5
Enter hours worked
10
Active or inactive?
active
Done. 1 rows inserted.

Content of Insurance Policy table

SSN | name | month | hour | active |
3 | Red | 5 | 10 | active |
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
```

```
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
3
Enter Volunteer SSN:
4
Enter date first join: 010112 - Jan 1, 2012
010100
Enter Recent Training Date
010111
Enter Recent Training Location
OK
Done. 1 rows inserted.

Content of Volunteer table

SSN | date_first join | date recent train | location |
3 | 10113 | 10119 | NV |
4 | 10100 | 10111 | OK |
Enter Team Associated With:
Red
Enter month worked
4
Enter hours worked
9
Active or inactive?
active
Done. 1 rows inserted.
```

```
Content of Insurance Policy table

SSN | name | month | hour | active |
3 | Red | 5 | 10 | active |
4 | Red | 4 | 9 | active |
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
```

```
20. Add a Person
21. Close
3
1. Enter Volunteer
2. Add Leader
2
Enter SSN:
3
Team name:
Red
Done. 1 rows inserted.
Leads Table
SSN: 3| d_name: Red|
```

<u>Query 4:</u>

Before Running:

```
INSERT INTO People VALUES
(1,'john',010100, 'white', 'm', 'dance', '11 robin rd', 'john@mail.com', 123456789, 234567890,
345678900, 'no')
INSERT INTO People VALUES
```

```
(2,'po',010110, 'asian', 'm', 'cs', '12 robinson rd', 'po@mail.com', 123456789, 234567890,
345678900, 'yes')
INSERT INTO People VALUES
(3,'cait',010205, 'white', 'f', 'ee', '1 king rd', 'cait@mail.com', 123445789, 2385676,
345678900, 'no')
INSERT INTO People VALUES
(4,'grace',010313, 'asian', 'f', 'music', '12 low rd', 'grace@mail.com', 123456789, 4582745,
6476583, 'yes')
INSERT INTO People VALUES
(5,'brand',010303, 'black', 'm', 'ame', '1 lowes rd', 'brand@mail.com', 123346789, 4672745,
6487583, 'yes'),
(6, 'sam', 010303, 'native', 'f', 'med', '12 lowes rd', 'sam@mail.com', 543346789, 4634745,
6457583, 'yes')

 INSERT INTO Team VALUES
('Yellow', 'medical', '010101'),
('Red', 'kitchen', '010102'),
('Blue', 'build', '010203')

INSERT INTO Serves VALUES
(3,'Red', 4, 10, 'active'),
(4, 'Red', 4, 9, 'active'),
```

Since Query 3 associated Team with Volunteers and entered hours, Query 4 will just enter more months that the volunteer or more teams a volunteer serves. So this Query will add the Serves table. I will highlight the Serve table.

Result:

```
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
```

```
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
4
Enter Volunteer SSN:
3
Enter Team Associated With:
Red
Enter month worked
5
Enter hours worked
10
Active or inactive?
active
Done. 1 rows inserted.
```

Content of Insurance Policy table

```
SSN | name | month | hour | active |
3 | Red | 4 | 10 | active |
3 | Red | 5 | 10 | active |
4 | Red | 4 | 9 | active |
```

Query 5

Before running:

```
INSERT INTO People VALUES
(1,'john',010100, 'white', 'm', 'dance', '11 robin rd', 'john@mail.com', 123456789, 234567890,
345678900, 'no')
INSERT INTO People VALUES
```

```
(2,'po',010110, 'asian', 'm', 'cs', '12 robinson rd', 'po@mail.com', 123456789, 234567890,
345678900, 'yes')
INSERT INTO People VALUES
(3,'cait',010205, 'white', 'f', 'ee', '1 king rd', 'cait@mail.com', 123445789, 2385676,
345678900, 'no')
INSERT INTO People VALUES
(4,'grace',010313, 'asian', 'f', 'music', '12 low rd', 'grace@mail.com', 123456789, 4582745,
6476583, 'yes')
INSERT INTO People VALUES
(5,'brand',010303, 'black', 'm', 'ame', '1 lowes rd', 'brand@mail.com', 123346789, 4672745,
6487583, 'yes'),
(6, 'sam', 010303, 'native', 'f', 'med', '12 lowes rd', 'sam@mail.com', 543346789, 4634745,
6457583, 'yes')

 INSERT INTO Team VALUES
('Yellow', 'medical', '010101'),
('Red', 'kitchen', '010102'),
('Blue', 'build', '010203')
```

Result:

```
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted by name
```

```
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
5
1. Enter Employee
2. Associated with more Team
1
Enter Employee SSN:
2
Enter salary:
458
Enter martial status
single
Enter hired date: 010112 - Jan 1, 2012
010110
Done. 1 rows inserted.

Content of Employee table

SSN | salary | martial status | hire date |
2 | 458 | single | 10110 |
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
```

```
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
5
1. Enter Employee
2. Associated with more Team
1
Enter Employee SSN:
3
Enter salary:
```

```
345
Enter martial status
single
Enter hired date: 010112 - Jan 1, 2012
030514
Done. 1 rows inserted.

Content of Employee table

SSN | salary | martial status | hire date |
2 | 458 | single | 10110 |
3 | 345 | single | 30514 |
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain anonymous
```

```
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
5
1. Enter Employee
2. Associated with more Team
2
Enter Employee SSN:
2
Enter team associated with
Yellow
Enter report description
stuff happened
Enter report date
010101
Done. 1 rows inserted.
```

Content of Reporting table

```
SSN | team | date | description |
2 | Yellow | 10101 | stuff happened |
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
```

```
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
5
1. Enter Employee
2. Associated with more Team
2
Enter Employee SSN:
2
Enter team associated with
Red
Enter report description
good food
Enter report date
```

```
010201
Done. 1 rows inserted.

Content of Reporting table

SSN | team | date | description |
2 | Red | 10201 | good food |
2 | Yellow | 10101 | stuff happened |
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
```

```
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
5
1. Enter Employee
2. Associated with more Team
2
Enter Employee SSN:
3
Enter team associated with
Blue
Enter report description
need more tools
Enter report date
010201
Done. 1 rows inserted.
```

Content of Reporting table

```
SSN | team | date | description |
3 | Blue | 10201 | need more tools |
2 | Red | 10201 | good food |
2 | Yellow | 10101 | stuff happened |
```

Query 6:

Before running:

```
INSERT INTO People VALUES
(1,'john',010100, 'white', 'm', 'dance', '11 robin rd', 'john@mail.com', 123456789, 234567890,
345678900, 'no')
INSERT INTO People VALUES
```

```sql
(2,'po',010110, 'asian', 'm', 'cs', '12 robinson rd', 'po@mail.com', 123456789, 234567890,
345678900, 'yes')
INSERT INTO People VALUES
(3,'cait',010205, 'white', 'f', 'ee', '1 king rd', 'cait@mail.com', 123445789, 2385676,
345678900, 'no')
INSERT INTO People VALUES
(4,'grace',010313, 'asian', 'f', 'music', '12 low rd', 'grace@mail.com', 123456789, 4582745,
6476583, 'yes')
INSERT INTO People VALUES
(5,'brand',010303, 'black', 'm', 'ame', '1 lowes rd', 'brand@mail.com', 123346789, 4672745,
6487583, 'yes'),
(6, 'sam', 010303, 'native', 'f', 'med', '12 lowes rd', 'sam@mail.com', 543346789, 4634745,
6457583, 'yes')

INSERT INTO Employee VALUES
(2, 458, 'single', 030320),
(3, 345, 'single', 030514)
```

Result:

```
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
```

```
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
6
Enter Employee SSN:
2
Enter date:
010516
Enter amount:
345
Enter description:
burger
Done. 1 rows inserted.

Content of Expense table

SSN | date | Amount| description |
2 | 10516 | 345 | burger |
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
```

```
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
6
Enter Employee SSN:
3
Enter date:
010115
Enter amount:
175
Enter description:
drinks
Done. 1 rows inserted.
```

Content of Expense table

```
SSN | date | Amount| description |
2 | 10516 | 345 | burger |
3 | 10115 | 175 | drinks |
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
```

```
19. Export
20. Add a Person
21. Close
6
Enter Employee SSN:
3
Enter date:
010115
Enter amount:
200
Enter description:
kooool-aid
Done. 1 rows inserted.
```

Content of Expense table

```
SSN | date | Amount| description |
2 | 10516 | 345 | burger |
3 | 10115 | 175 | drinks |
3 | 10115 | 200 | kooool-aid |
```

Query 7:

Before Running:

```
 INSERT INTO Team VALUES
('Yellow', 'medical', '010101'),
('Red', 'kitchen', '010102')
```

Result:

```
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
7
```

```
1. Enter a new organization
2. Associate with team
1
Enter Orgazniation's Name:
SKT T1
Enter Address:
1 kor dr
Enter phone number:
3452304
Enter contact People
le
Done. 1 rows inserted.

Content of Org table

Name | address | phone| contact |
SKT T1 | 1 kor dr | 3452304 | le |
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
```

```
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
7
1. Enter a new organization
2. Associate with team
1
Enter Orgazniation's Name:
GENG
Enter Address:
2 word dr
Enter phone number:
4323444
Enter contact People
sophie
Done. 1 rows inserted.

Content of Org table

Name | address | phone| contact |
GENG | 2 word dr | 4323444 | sophie |
SKT T1 | 1 kor dr | 3452304 | le |
Enter you option(1-17):
```

```
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
7
1. Enter a new organization
2. Associate with team
2
```

```
Enter Orgazniation Name:
SKT T1
Enter Team Name:
Red
Done. 1 rows inserted.

Content of Sponsor table

Org | Team |
SKT T1 | Red |
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
```

```
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
7
1. Enter a new organization
2. Associate with team
2
Enter Orgazniation Name:
GENG
Enter Team Name:
Yellow
Done. 1 rows inserted.

Content of Sponsor table

Org | Team |
GENG | Yellow |
SKT T1 | Red |
```

Query 8:

Before running:

```
INSERT INTO People VALUES
(2,'po',010110, 'asian', 'm', 'cs', '12 robinson rd', 'po@mail.com', 123456789, 234567890,
345678900, 'yes')
```

```sql
INSERT INTO People VALUES
(3,'cait',010205, 'white', 'f', 'ee', '1 king rd', 'cait@mail.com', 123445789, 2385676,
345678900, 'no')
INSERT INTO People VALUES
(4,'grace',010313, 'asian', 'f', 'music', '12 low rd', 'grace@mail.com', 123456789, 4582745,
6476583, 'yes')
INSERT INTO People VALUES
(5,'brand',010303, 'black', 'm', 'ame', '1 lowes rd', 'brand@mail.com', 123346789, 4672745,
6487583, 'yes'),
(6, 'sam', 010303, 'native', 'f', 'med', '12 lowes rd', 'sam@mail.com', 543346789, 4634745,
6457583, 'yes')
```

Result:

```
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
```

```
19. Export
20. Add a Person
21. Close
8
1. Enter a new donor
2. Associate donor with 1 or more Donation(s)
1
Enter SSN:
1
Is it anonymous
yes
Done. 1 rows inserted.

Content of Donor table

SSN | anonymous |
1 | yes |
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
```

```
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
8
1. Enter a new donor
2. Associate donor with 1 or more Donation(s)
1
Enter SSN:
2
Is it anonymous
yes
Done. 1 rows inserted.

Content of Donor table

SSN | anonymous |
1 | yes |
2 | yes |
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
```

```
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
8
1. Enter a new donor
2. Associate donor with 1 or more Donation(s)
2
Donor SSN:
1
Date:
010115
Amount
34
Campaign:
fdss
Type:
1. Card
2. Check
2
Check number:
54332
Done. 1 rows inserted.
```

Content of Donate table

```
SSN | date | amount|type|camp
```

```
1 | 10115 |34 | check |fdss |
Done. 1 rows inserted.

Content of Donor Check table

SSN | date | amount | check num|
1 | 10115 |34 | 54332 |
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
8
```

```
1. Enter a new donor
2. Associate donor with 1 or more Donation(s)
2
Donor SSN:
1
Date:
010315
Amount
35
Campaign:
tre
Type:
1. Card
2. Check
1
Card number:
2345
Card type:
master
Exp date: 010112 - Jan 1, 2012
010120
Done. 1 rows inserted.
Done. 1 rows inserted.
```

Content of Donor Card table

```
SSN | date | amount|card num|card type| exp date|
1 | 10315 | 35 | 2345 | master | 10120 |
```

<u>Query 9:</u>

Before running:

```
INSERT INTO External_Organization VALUES
('SKT T1', '1 kor dr',3452304, 'le'),
('GENG', '2 word dr', 4323444, 'sophie')
```

Result:

```
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
9
1. Enter new org that donates
2. Associated with 1 or more Donation(s)
1
Enter Org name:
SKT T1
```

```
Is it anonymous
yes
Done. 1 rows inserted.

Content of Donor Card table

Org Name | anonymous |
SKT T1 | yes |
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
```

```
9
1. Enter new org that donates
2. Associated with 1 or more Donation(s)
2
Org Name:
SKT T1
Date:
010115
Amount
34
Campaign:
fdss
Type:
1. Card
2. Check
2
Check number:
54332
Done. 1 rows inserted.

Content of Donor Card table

Org Name | anonymous |
SKT T1 | yes |
Done. 1 rows inserted.

Content of Org Card table

SSN | date | amount | check num|
SKT T1 | 10115 |34 | 54332 |
```

Query 10:

Before running:

```
INSERT INTO People VALUES
(1,'john',010100, 'white', 'm', 'dance', '11 robin rd', 'john@mail.com', 123456789, 234567890,
345678900, 'no')
INSERT INTO People VALUES
```

```
(2,'po',010110, 'asian', 'm', 'cs', '11 robin rd', 'john@mail.com', 123456789, 234567890,
345678900, 'yes')
INSERT INTO Client VALUES
INSERT INTO Client VALUES
(1, 'ben', 1234567, 'son', 3456789, 0101204),
(2, 'john', 5768496, 'cow', 5869485, 010204)
```

```
Result:
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be
sorted by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15.
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Quit

10
Enter SSN of Client:
2
SSN: 2|Doctor name: john | Phone 5768496 |
(1,'john',010100, 'white', 'm', 'dance', '11 robin rd', 'john@mail.com', 123456789, 234567890,
345678900, 'no')
```

Query 11:

Before running:

```
INSERT INTO People VALUES
(1,'john',010100, 'white', 'm', 'dance', '11 robin rd', 'john@mail.com', 123456789, 234567890,
345678900, 'no')
INSERT INTO People VALUES
```

```
(2,'po',010110, 'asian', 'm', 'cs', '12 robinson rd', 'po@mail.com', 123456789, 234567890,
345678900, 'yes')
INSERT INTO People VALUES
(3,'cait',010205, 'white', 'm', 'ee', '1 king rd', 'cait@mail.com', 123445789, 2385676,
345678900, 'no')
INSERT INTO Employee VALUES
(1, 500, 'single', 010110),
(2, 458, 'married', 030320),
(3, 345, 'single', 030514)

INSERT INTO Expenses VALUES
(1, 010115, 350, 'ice cream'),
(1, 010115, 10, 'water'),
(2, 010516, 345, 'burger'),
(3, 010115, 175, 'drinks')
(3, 010115, 200, 'kooool-aid')
```

Result:

```
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
```

```
15.
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Quit

11
Expense date:
010115
SSN: 1|Amount: 450 |
SSN: 3|Amount: 375 |
```

Query 12:

Before running:

```
INSERT INTO People VALUES
(1,'john',010100, 'white', 'm', 'dance', '11 robin rd', 'john@mail.com', 123456789, 234567890,
345678900, 'no')
INSERT INTO People VALUES
```

```
(2,'po',010110, 'asian', 'm', 'cs', '12 robinson rd', 'po@mail.com', 123456789, 234567890,
345678900, 'yes')
INSERT INTO People VALUES
(3,'cait',010205, 'white', 'f', 'ee', '1 king rd', 'cait@mail.com', 123445789, 2385676,
345678900, 'no')
INSERT INTO People VALUES
(4,'grace',010313, 'asian', 'f', 'music', '12 low rd', 'grace@mail.com', 123456789, 4582745,
6476583, 'yes')
INSERT INTO People VALUES
(5,'brand',010303, 'black', 'm', 'ame', '1 lowes rd', 'brand@mail.com', 123346789, 4672745,
6487583, 'yes')

INSERT INTO Team VALUES
('Yellow', 'medical', 010101),
('Red', 'kitchen', 010102)

INSERT INTO Client VALUES
(1, 'ben', 1234567, 'son', 3456789, 0101204),
(2, 'john', 5768496, 'cow', 5869485, 010204)

INSERT INTO Volunteers VALUES
(3, 010113, 010119, 'NV'),
(4,01010, 010111, 'OK'),
(5, 01011, 010112, 'TX')

INSERT INTO Serves VALUES
(3,'Red', 4, 10, 'active'),
(3,'Red', 5, 10, 'active'),
(4, 'Red', 4, 9, 'active'),
(5,'Yellow', 3, 11, 'active')

INSERT INTO Cares VALUES
(1,'Yellow','active'),
(2,'Red', 'active')
```

Result:

```
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
```

```
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Quit

12
Enter client's SSN
2
SSN: 3|
SSN: 4|
```

Query 13:

Before running:

```sql
INSERT INTO People VALUES
(1,'john',010100, 'white', 'm', 'dance', '11 robin rd', 'john@mail.com', 123456789, 234567890,
345678900, 'no')
INSERT INTO People VALUES
```

```
(2,'po',010110, 'asian', 'm', 'cs', '12 robinson rd', 'po@mail.com', 123456789, 234567890,
345678900, 'yes')
INSERT INTO People VALUES
(3,'cait',010205, 'white', 'f', 'ee', '1 king rd', 'cait@mail.com', 123445789, 2385676,
345678900, 'no')
INSERT INTO People VALUES
(4,'grace',010313, 'asian', 'f', 'music', '12 low rd', 'grace@mail.com', 123456789, 4582745,
6476583, 'yes')
INSERT INTO People VALUES
(5,'brand',010303, 'black', 'm', 'ame', '1 lowes rd', 'brand@mail.com', 123346789, 4672745,
6487583, 'yes'),
(6, 'sam', 010303, 'native', 'f', 'med', '12 lowes rd', 'sam@mail.com', 543346789, 4634745,
6457583, 'yes')

INSERT INTO Team VALUES
('Yellow', 'medical', 010101),
('Red', 'kitchen', 010102)

INSERT INTO Client VALUES
(1, 'ben', 1234567, 'son', 3456789, 0101204),
(2, 'john', 5768496, 'cow', 5869485, 010204),
(6, 'bo', 432244, 'cao', 432424,01010)

INSERT INTO Volunteers VALUES
(3, 010113, 010119, 'NV'),
(4,01010, 010111, 'OK'),
(5, 01011, 010112, 'TX')

INSERT INTO Serves VALUES
(3,'Red', 4, 10, 'active'),
(3,'Red', 5, 10, 'active'),
(4, 'Red', 4, 9, 'active'),
(5,'Yellow', 3, 11, 'active')

INSERT INTO Cares VALUES
(1,'Yellow','active'),
(2,'Red', 'active'),
(6, 'Yellow', 'active')

INSERT INTO External_Organization VALUES
('SKT T1', '1 kor dr',3452304, 'le'),
('GENG', '2 word dr', 4323444, 'sophie')

INSERT INTO Sponsor VALUES
('SKT T1', 'Red'),
('GENG', 'Yellow')
```

Result:

```
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Quit

13
name: john mail: 11 robin rd email: john@mail.com home: 123456789 work: 234567890 cell:
345678900
name: sam mail: 12 lowes rd email: sam@mail.com home: 543346789 work: 4634745 cell:
6457583
```

Query 14:

Before running:

```
INSERT INTO People VALUES
(1,'john',010100, 'white', 'm', 'dance', '11 robin rd', 'john@mail.com', 123456789, 234567890,
345678900, 'no')
INSERT INTO People VALUES
```

```
(2,'po',010110, 'asian', 'm', 'cs', '12 robinson rd', 'po@mail.com', 123456789, 234567890,
345678900, 'yes')
INSERT INTO People VALUES
(3,'cait',010205, 'white', 'f', 'ee', '1 king rd', 'cait@mail.com', 123445789, 2385676,
345678900, 'no')
INSERT INTO People VALUES
(4,'grace',010313, 'asian', 'f', 'music', '12 low rd', 'grace@mail.com', 123456789, 4582745,
6476583, 'yes')
INSERT INTO People VALUES
(5,'brand',010303, 'black', 'm', 'ame', '1 lowes rd', 'brand@mail.com', 123346789, 4672745,
6487583, 'yes'),
(6, 'sam', 010303, 'native', 'f', 'med', '12 lowes rd', 'sam@mail.com', 543346789, 4634745,
6457583, 'yes')




INSERT INTO Donor VALUES
(1, 'yes')
INSERT INTO Donor VALUES
(2, 'yes')

INSERT INTO Donor VALUES
(6, 'yes')

INSERT INTO Donor_Donate VALUES (1,010115,34, 'card', 'fdss')
INSERT INTO Donor_Donate VALUES (1,010315,35, 'card', 'tre')
INSERT INTO Donor_Donate VALUES (2,010116,35, 'check','ros')
INSERT INTO Donor_Donate VALUES (2,050116,36, 'check','ros')
INSERT INTO Donor_Donate VALUES (6,030116,30, 'check','ros')
INSERT INTO Donor_Donate_Check VALUES (2,010116,35, 54332)




INSERT INTO Employee VALUES
(1, 500, 'single', 010110),
(2, 458, 'married', 030320),
(3, 345, 'single', 030514)
```

Result:

Enter you option(1-17):

1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team

```
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Quit

14
name: john| total amount: 69| anonymous: yes
name: po| total amount: 71| anonymous: yes
```

Query 15:

Before running:

```
INSERT INTO Team VALUES
('Yellow', 'medical', '010101'),
('Red', 'kitchen', '010102')
```

```
Enter you option(1-17):
1. Enter a new team into the database
```

```
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Quit

15
Enter date:
010201
name: Red|
```

Query 16

Before running:

```
INSERT INTO People VALUES
(1,'john',010100, 'white', 'm', 'dance', '11 robin rd', 'john@mail.com', 123456789, 234567890,
345678900, 'no')
INSERT INTO People VALUES
```

```
(2,'po',010110, 'asian', 'm', 'cs', '12 robinson rd', 'po@mail.com', 123456789, 234567890,
345678900, 'yes')
INSERT INTO People VALUES
(3,'cait',010205, 'white', 'f', 'ee', '1 king rd', 'cait@mail.com', 123445789, 2385676,
345678900, 'no')
INSERT INTO People VALUES
(4,'grace',010313, 'asian', 'f', 'music', '12 low rd', 'grace@mail.com', 123456789, 4582745,
6476583, 'yes')
INSERT INTO People VALUES
(5,'brand',010303, 'black', 'm', 'ame', '1 lowes rd', 'brand@mail.com', 123346789, 4672745,
6487583, 'yes'),
(6, 'sam', 010303, 'native', 'f', 'med', '12 lowes rd', 'sam@mail.com', 543346789, 4634745,
6457583, 'yes')

INSERT INTO Employee VALUES
(1, 500, 'single', 010110),
(2, 458, 'married', 030320),
(3, 345, 'single', 030514)

 INSERT INTO Team VALUES
('Yellow', 'medical', '010101'),
('Red', 'kitchen', '010102'),
('Blue', 'build', '010203')

INSERT INTO Reporting VALUES
(2, 'Yellow', 010101, 'stuff happened'),
(2, 'Red', 010201,'good food'),
(3, 'Blue', 010301, 'need more tools')
```

Result:

```
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
```

12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams sponsored by an organization whose name starts with a letter between B and K. The client list should be sorted by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list should be sorted by the total amount of the donations, and indicate if each donor wishes to remain anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for transportation is less than 5
18. Close

16

```
SSN: 1| Salary: 500|
SSN: 2| Salary: 503|
SSN: 3| Salary: 345|
```

Query 17:

Before running:

```
INSERT INTO People VALUES
(1,'john',010100, 'white', 'm', 'dance', '11 robin rd', 'john@mail.com', 123456789, 234567890,
345678900, 'no')
INSERT INTO People VALUES
```

```
(2,'po',010110, 'asian', 'm', 'cs', '12 robinson rd', 'po@mail.com', 123456789, 234567890,
345678900, 'yes')
INSERT INTO People VALUES
(3,'cait',010205, 'white', 'f', 'ee', '1 king rd', 'cait@mail.com', 123445789, 2385676,
345678900, 'no')
INSERT INTO People VALUES
(4,'grace',010313, 'asian', 'f', 'music', '12 low rd', 'grace@mail.com', 123456789, 4582745,
6476583, 'yes')
INSERT INTO People VALUES
(5,'brand',010303, 'black', 'm', 'ame', '1 lowes rd', 'brand@mail.com', 123346789, 4672745,
6487583, 'yes'),
(6, 'sam', 010303, 'native', 'f', 'med', '12 lowes rd', 'sam@mail.com', 543346789, 4634745,
6457583, 'yes')

INSERT INTO Client VALUES
(1, 'ben', 1234567, 'son', 3456789, 0101204),
(2, 'john', 5768496, 'cow', 5869485, 010204),
(6, 'bo', 432244, 'cao', 432424,01010)

INSERT INTO Insurance_Policy VALUES
(1,34, 43,'ad1', 'Auto'),
(2,45, 54, 'ad2', 'Health'),
(2,67, 76, 'ad2', 'Home'),
(6,43, 34, 'ad6', 'Life'),
(6,54, 54, 'ad6', 'Auto')

INSERT INTO Needs VALUES
(1, 'Transportation', 6),
(1, 'Visiting', 6),
(2, 'Transportation', 9),
(6, 'Transportation', 4),
(6, 'Food', 6)
```

Result:

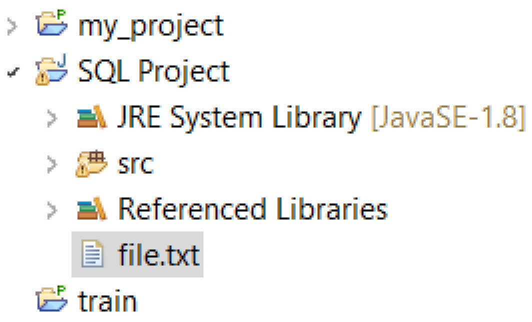
```
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
```

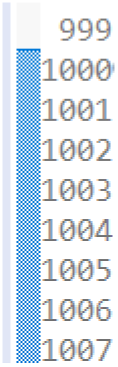
```
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
17
SSN: 1| d_name: ben| d_phone: 1234567| a_name: son| a_name: 3456789| type: 3456789|
SSN: 2| d_name: john| d_phone: 5768496| a_name: cow| a_name: 5869485| type: 5869485|
```

Query 18:

File.txt:

```
Black,tech,010100
White,plan,010205
Orange,trash,040502
```





```
999
1000
1001
1002
1003
1004
1005
1006
1007
```

Result:

```
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
```

```
21. Close
18
Name: Black| type: tech| date: 10100|
Name: Orange| type: trash| date: 40502|
Name: White| type: plan| date: 10205|
```

Query 19:

Before running:

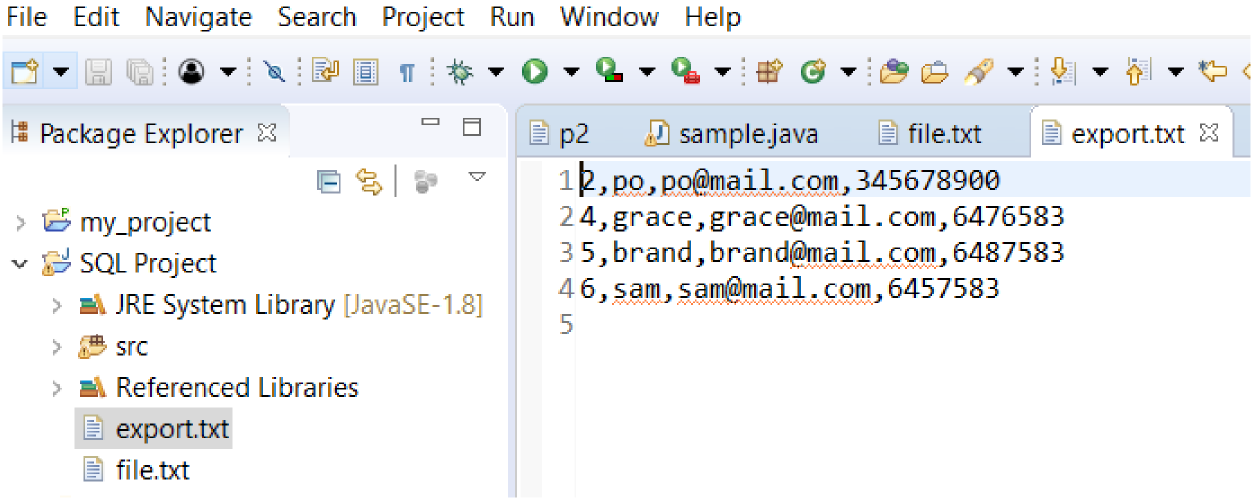
```
INSERT INTO People VALUES
(1,'john',010100, 'white', 'm', 'dance', '11 robin rd', 'john@mail.com', 123456789, 234567890,
345678900, 'no')
INSERT INTO People VALUES
```

```sql
(2,'po',010110, 'asian', 'm', 'cs', '12 robinson rd', 'po@mail.com', 123456789, 234567890,
345678900, 'yes')
INSERT INTO People VALUES
(3,'cait',010205, 'white', 'f', 'ee', '1 king rd', 'cait@mail.com', 123445789, 2385676,
345678900, 'no')
INSERT INTO People VALUES
(4,'grace',010313, 'asian', 'f', 'music', '12 low rd', 'grace@mail.com', 123456789, 4582745,
6476583, 'yes')
INSERT INTO People VALUES
(5,'brand',010303, 'black', 'm', 'ame', '1 lowes rd', 'brand@mail.com', 123346789, 4672745,
6487583, 'yes'),
(6, 'sam', 010303, 'native', 'f', 'med', '12 lowes rd', 'sam@mail.com', 543346789, 4634745,
6457583, 'yes')
```

Result:

```
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
```

```
18. Import
19. Export
20. Add a Person
21. Close
19
Write in file export
Write in file export
Write in file export
Write in file export
```



```
2,po,po@mail.com,345678900
4,grace,grace@mail.com,6476583
5,brand,brand@mail.com,6487583
6,sam,sam@mail.com,6457583
```

Query 20:

```
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
20
Enter SSN
1
Enter name
john
Enter date of birth: 010101-Jan 1,2001
010100
```

```
Enter race
white
Enter gender
m
Enter profession
dance
Enter mail address
11 robin rd
Enter email address
john@mail.com
Enter home phone number
123456789
Enter work phone number
234567890
Enter Cell phone number
345678900
On mailling list?
no
Done. 1 rows inserted.
SSN: 1| name: john| dob: 10100| race: white| gender: m| prof: dance|mail: 11 robin rd| email:
john@mail.com| home: 123456789| work: 234567890| cel: 345678900|
------------------------------------------------------------------------------------------------------------
```

After this, you add an External Org using Query 7. For the sake of not running more lines, this is the content of table for External Org in this example:

```
INSERT INTO External_Organization VALUES
('SKT T1', '1 kor dr',3452304, 'le')
```

```
------------------------------------------------------------------------------------------------------------
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
```

```
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
20
1. Enter Person
2. Affiliate with an Org
2
Enter SSN:
1
Enter Org name
SKT T1
Done. 1 rows inserted.
SSN: 1| org: SKT T1|
```

Query 21

```
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
20
Later!
```

Error 1:

- Duplication: (Team name)

```
------------------------------------------------------------------------------------------------------------------
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
1
Please enter team name:
SKT
Please enter team type:
clean
```

```
Please enter date formed: 010112 = Jan 1, 2012
010112
Done. 1 rows inserted.

Content of Team table

Team Name | Team Type | Date Formed
SKT | clean | 010112 |
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
```

```
1
Please enter team name:
SKT
Please enter team type:
housing
Please enter date formed: 010112 = Jan 1, 2012
010101
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: Violation of
PRIMARY KEY constraint 'PK__Team__B85BEBFF2648DAF1'. Cannot insert duplicate key in
object 'dbo.Team'. The duplicate key value is (SKT).
	at
com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDatabaseError(SQLServerExcepti
on.java:262)
	at
com.microsoft.sqlserver.jdbc.SQLServerStatement.getNextResult(SQLServerStatement.java:163
2)
	at
com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.doExecutePreparedStatement(SQLS
erverPreparedStatement.java:602)
	at
com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement$PrepStmtExecCmd.doExecute(SQL
ServerPreparedStatement.java:524)
	at com.microsoft.sqlserver.jdbc.TDSCommand.execute(IOBuffer.java:7375)
	at
com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCommand(SQLServerConnection.ja
va:3206)
	at
com.microsoft.sqlserver.jdbc.SQLServerStatement.executeCommand(SQLServerStatement.java:
247)
	at
com.microsoft.sqlserver.jdbc.SQLServerStatement.executeStatement(SQLServerStatement.java:
222)
	at
com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.executeUpdate(SQLServerPrepared
Statement.java:473)
	at sample.Q1(sample.java:180)
	at sample.main(sample.java:90)
```

Error 2:

- Wrong data type: (Organization Phone Number)

```
-----------------------------------------------------------------------------------------------------------
Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be sorted
by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
```

```
7
1. Enter a new organization
2. Associate with team
3. Affiliate with a person
1
Enter Orgazniation's Name:
SKT
Enter Address:
11 kk
Enter phone number:
reewq
Exception in thread "main" java.lang.NumberFormatException: For input string: "reewq"
	at java.lang.NumberFormatException.forInputString(Unknown Source)
	at java.lang.Integer.parseInt(Unknown Source)
	at java.lang.Integer.parseInt(Unknown Source)
	at sample.Q7(sample.java:682)
	at sample.main(sample.java:108)
```

Error 3:

- Foreign key doesn’t exist: (Donor’s SSN)

Before running:

```
INSERT INTO People VALUES
(1,'john',010100, 'white', 'm', 'dance', '11 robin rd', 'john@mail.com', 123456789, 234567890,
345678900, 'no')
INSERT INTO People VALUES
(2,'po',010110, 'asian', 'm', 'cs', '12 robinson rd', 'po@mail.com', 123456789, 234567890,
345678900, 'yes')
INSERT INTO People VALUES
(3,'cait',010205, 'white', 'f', 'ee', '1 king rd', 'cait@mail.com', 123445789, 2385676,
345678900, 'no')
INSERT INTO People VALUES
(4,'grace',010313, 'asian', 'f', 'music', '12 low rd', 'grace@mail.com', 123456789, 4582745,
6476583, 'yes')
INSERT INTO People VALUES
```

```
(5,'brand',010303, 'black', 'm', 'ame', '1 lowes rd', 'brand@mail.com', 123346789, 4672745,
6487583, 'yes'),
(6, 'sam', 010303, 'native', 'f', 'med', '12 lowes rd', 'sam@mail.com', 543346789, 4634745,
6457583, 'yes')

Enter you option(1-17):
1. Enter a new team into the database
2. Enter a new client into the database and associate him or her with one or more teams
3. Enter a new volunteer into the database and associate him or her with one or more teams
4. Enter the number of hours a volunteer worked this month for a particular team
5. Enter a new employee into the database and associate him or her with one or more teams
6. Enter an expense charged by an employee
7. Enter a new organization and associate it to one or more PAN teams
8. Enter a new donor and associate him or her with several donations.
9. Enter a new organization and associate it with several donations
10. Retrieve the name and phone number of the doctor of a particular client
11. Retrieve the total amount of expenses charged by each employee for a particular period of
time.
The list should be sorted by the total amount of expenses
12. Retrieve the list of volunteers that are members of teams that support a particular client
13. Retrieve the names and contact information of the clients that are supported by teams
sponsored by
an organization whose name starts with a letter between B and K. The client list should be
sorted by name
14. Retrieve the name and total amount donated by Donor that are also employees. The list
should be
sorted by the total amount of the donations, and indicate if each donor wishes to remain
anonymous
15. Retrieve the names of all teams that were founded after a particular date(1/month)
16. Increase the salary by 10% of all employees to whom more than one team must report
17. Delete all clients who do not have health insurance and whose value of importance for
transportation
is less than 5
18. Import
19. Export
20. Add a Person
21. Close
8
1. Enter a new donor
2. Associate donor with 1 or more Donation(s)
1
Enter SSN:
7
Is it anonymous
yes
```

```
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: The INSERT
statement conflicted with the FOREIGN KEY constraint "FK__Donor__SSN__0A76A605". The conflict
occurred in database "cs-dsa-4513-sql-db", table "dbo.People", column 'SSN'.
        at
com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDatabaseError(SQLServerException.java:
262)
        at
com.microsoft.sqlserver.jdbc.SQLServerStatement.getNextResult(SQLServerStatement.java:1632)
        at
com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.doExecutePreparedStatement(SQLServerPr
eparedStatement.java:602)
        at
com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement$PrepStmtExecCmd.doExecute(SQLServerPre
paredStatement.java:524)
        at com.microsoft.sqlserver.jdbc.TDSCommand.execute(IOBuffer.java:7375)
        at
com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCommand(SQLServerConnection.java:3206)
        at
com.microsoft.sqlserver.jdbc.SQLServerStatement.executeCommand(SQLServerStatement.java:247)
        at
com.microsoft.sqlserver.jdbc.SQLServerStatement.executeStatement(SQLServerStatement.java:222)
        at
com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.executeUpdate(SQLServerPreparedStateme
nt.java:473)
        at sample.Q8(sample.java:739)
        at sample.main(sample.java:111)
```

Task 7:

SQL Code:

```
DROP TABLE IF EXISTS Team; --Drop the table if it was previously created

--Create the new table for movie_nights schedule
CREATE TABLE Teams (
    Name VARCHAR(64)  PRIMARY KEY ,
    Type VARCHAR(64),
    Date_formed DATETIME
);

--Insert two records to begin with
INSERT INTO Teams
VALUES
 ('Red','Clean','2019-12-31 20:00:00'),
 ('Yellow','Medical','2020-01-03 19:00:00');
```

JSP:

- Get table:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
  pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
  <meta charset="UTF-8">
    <title>Team</title>
  </head>
  <body>
    <%@page import="jsp_azure_test.DataHandler"%>
    <%@page import="java.sql.ResultSet"%>
    <%
      // We instantiate the data handler here, and get all the movies from the database
      final DataHandler handler = new DataHandler();
      final ResultSet movies = handler.getAllMovies();
    %>
    <!-- The table for displaying all the movie records -->
    <table cellspacing="2" cellpadding="2" border="1">
      <tr> <!-- The table headers row -->
       <td align="center">
        <h4>Team Name</h4>
       </td>
       <td align="center">
        <h4>Type</h4>
       </td>
       <td align="center">
        <h4>Date Formed</h4>
       </td>
      </tr>
      <%
       while(movies.next()) { // For each movie_night record returned...
          // Extract the attribute values for every row returned
          final String Name = movies.getString("Name");
          final String Type = movies.getString("Type");
          final String Date_formed = movies.getString("Date_formed");

          out.println("<tr>"); // Start printing out the new table row
          out.println( // Print each attribute value
            "<td align=\"center\">" + Name +
            "</td><td align=\"center\"> " + Type +
            "</td><td align=\"center\"> " + Date_formed + "</td>");
          out.println("</tr>");
        }
        %>
     </table>
  </body>
</html>
```

- Insert into table form

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Add Movie Night</title>
  </head>
  <body>
    <h2>Add Movie Night</h2>
    <!--
      Form for collecting user input for the new movie_night record.
      Upon form submission, add_movie.jsp file will be invoked.
    -->
    <form action="add_movie.jsp">
      <!-- The form organized in an HTML table for better clarity. -->
      <table border=1>
        <tr>
          <th colspan="2">Enter the Team info:</th>
        </tr>
        <tr>
          <td>Name:</td>
          <td><div style="text-align: center;">
          <input type=text name=Name>
          </div></td>
        </tr>
        <tr>
          <td>Type:</td>
          <td><div style="text-align: center;">
          <input type=text name=Type>
          </div></td>
```

```
            </tr>
            <tr>
               <td>Date_formed:</td>
               <td><div style="text-align: center;">
               <input type=text name=Date_formed>
               </div></td>
            </tr>
            <tr>
               <td><div style="text-align: center;">
               <input type=reset value=Clear>
               </div></td>
               <td><div style="text-align: center;">
               <input type=submit value=Insert>
               </div></td>
            </tr>
         </table>
      </form>
   </body>
</html>
```

- Insert into table action:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Query Result</title>
</head>
   <body>
   <%@page import="jsp_azure_test.DataHandler"%>
   <%@page import="java.sql.ResultSet"%>
   <%@page import="java.sql.Array"%>
   <%
   // The handler is the one in charge of establishing the connection.
   DataHandler handler = new DataHandler();

   // Get the attribute values passed from the input form.
   String Name = request.getParameter("Name");
```

```
  String Type = request.getParameter("Type");
  String Date_formed = request.getParameter("Date_formed");


  /*
   * If the user hasn't filled out all the time, movie name and duration. This is very simple checking.
   */
  if (Name.equals("") || Type.equals("") || Date_formed.equals("")) {
    response.sendRedirect("add_movie_form.jsp");
  } else {

    // Now perform the query with the data from the form.
    boolean success = handler.addMovie(Name, Type, Date_formed);
    if (!success) { // Something went wrong
      %>
        <h2>There was a problem inserting the course</h2>
      <%
    } else { // Confirm success to the user
      %>
      <h2>Team:</h2>

      <ul>
        <li>Name: <%=Name%></li>
        <li>Type: <%=Type%></li>
        <li>Date_formed: <%=Date_formed%></li>
      </ul>

      <h2>Was successfully inserted.</h2>

      <a href="get_all_movies.jsp">See all Teams..</a>
      <%
    }
  }
  %>
  </body>
</html>
```

- Date form:

```
<!DOCTYPE html>
```

```
<html>
   <head>
      <meta charset="UTF-8">
      <title>Date After</title>
   </head>
   <body>
      <h2>Insert Date</h2>
      <!--
         Form for collecting user input for the new movie_night record.
         Upon form submission, add_movie.jsp file will be invoked.
      -->
      <form action="AfterDate.jsp">
         <!-- The form organized in an HTML table for better clarity. -->
         <table border=1>
            <tr>
               <th colspan="2">Enter the Date:</th>
            </tr>
            <tr>
               <td>Date:</td>
               <td><div style="text-align: center;">
               <input type=text name=Date>
               </div></td>
            </tr>
            <tr>
               <td><div style="text-align: center;">
               <input type=reset value=Clear>
               </div></td>
               <td><div style="text-align: center;">
               <input type=submit value=Insert>
               </div></td>
            </tr>
         </table>
      </form>
   </body>
</html>
```

- Date Action:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
```

```
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Query Result</title>
</head>
    <body>
    <%@page import="jsp_azure_test.DataHandler"%>
    <%@page import="java.sql.ResultSet"%>
    <%@page import="java.sql.Array"%>
    <%
    // The handler is the one in charge of establishing the connection.
    DataHandler handler = new DataHandler();

    // Get the attribute values passed from the input form.
    String Date = request.getParameter("Date");
    // function get Team from DataHandler
    final ResultSet movies = handler.getTeams(Date);
        %>
        <table cellspacing="2" cellpadding="2" border="1">
            <tr> <!-- The table headers row -->
              <td align="center">
                <h4>Team Name</h4>
              </td>
              <td align="center">
                <h4>Type</h4>
              </td>
              <td align="center">
                <h4>Date Formed</h4>
              </td>

            </tr>
    <%
    /*
     * If the user hasn't filled out all the time, movie name and duration. This is very simple
checking.
     */
    if (Date.equals("")) {
        response.sendRedirect("DateAfter_form.jsp");
    } else {
        while(movies.next()) { // For each movie_night record returned...
            // Extract the attribute values for every row returned
            final String Name = movies.getString("Name");
            final String Type = movies.getString("Type");
            final String Date_formed = movies.getString("Date_formed");

            out.println("<tr>"); // Start printing out the new table row
            out.println( // Print each attribute value
                "<td align=\"center\">" + Name +
                "</td><td align=\"center\"> " + Type +
                "</td><td align=\"center\"> " + Date_formed + "</td>");
            out.println("</tr>");
            }
        }

    %>
    </body>
</html>
```

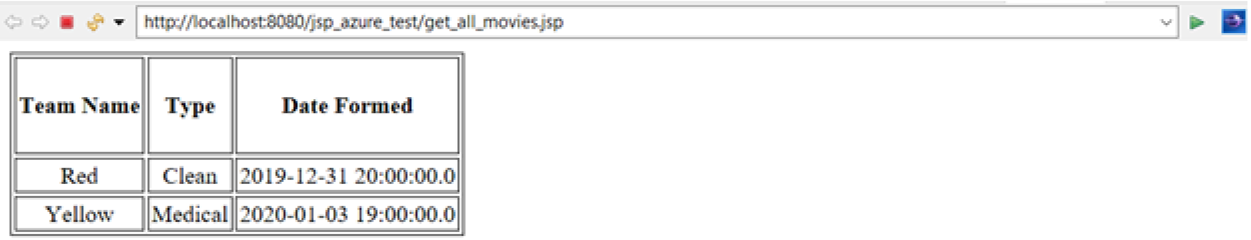
Java:

```
package jsp_azure_test;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

public class DataHandler {

   private Connection conn;

   // Azure SQL connection credentials
   private String server = "<username>-sql-server.database.windows.net";
   private String database = "cs-dsa-4513-sql-db";
   private String username = "<username>";
   private String password = "<password>";

   // Resulting connection string
   final private String url =

String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;trust
ServerCertificate=false;hostNameInCertificate=*.database.windows.net;loginTimeout=30;",
               server, database, username, password);

   // Initialize and save the database connection
   private void getDBConnection() throws SQLException {
      if (conn != null) {
         return;
      }

      this.conn = DriverManager.getConnection(url);
   }

   // Return the result of selecting everything from the movie_night table
   public ResultSet getAllMovies() throws SQLException {
      getDBConnection();
```

```
    final String sqlQuery = "SELECT * FROM Teams;";
    final PreparedStatement stmt = conn.prepareStatement(sqlQuery);
    return stmt.executeQuery();
  }

  // Inserts a record into the movie_night table with the given attribute values
  public boolean addMovie(
      String Name, String Type, String Date_formed) throws SQLException {

    getDBConnection(); // Prepare the database connection

    // Prepare the SQL statement
    final String sqlQuery =
        "INSERT INTO Teams " +
        "VALUES " +
        "(?, ?, ?)";
    final PreparedStatement stmt = conn.prepareStatement(sqlQuery);

    // Replace the '?' in the above statement with the given attribute values
    stmt.setString(1, Name);
    stmt.setString(2, Type);
    stmt.setString(3, Date_formed);
    // Execute the query, if only one record is updated, then we indicate success by returning
true
    return stmt.executeUpdate() == 1;
  }
  public ResultSet getTeams(String Date) throws SQLException {

    getDBConnection(); // Prepare the database connection

    // Prepare the SQL statement
    final String sqlQuery =
        "SELECT * FROM Teams WHERE Date_formed > Convert(DateTime, '"+Date+"' )";
    final PreparedStatement stmt = conn.prepareStatement(sqlQuery);

    // Replace the '?' in the above statement with the given attribute values
    // Execute the query, if only one record is updated, then we indicate success by returning
true
    return stmt.executeQuery();
  }}
```

Execution:

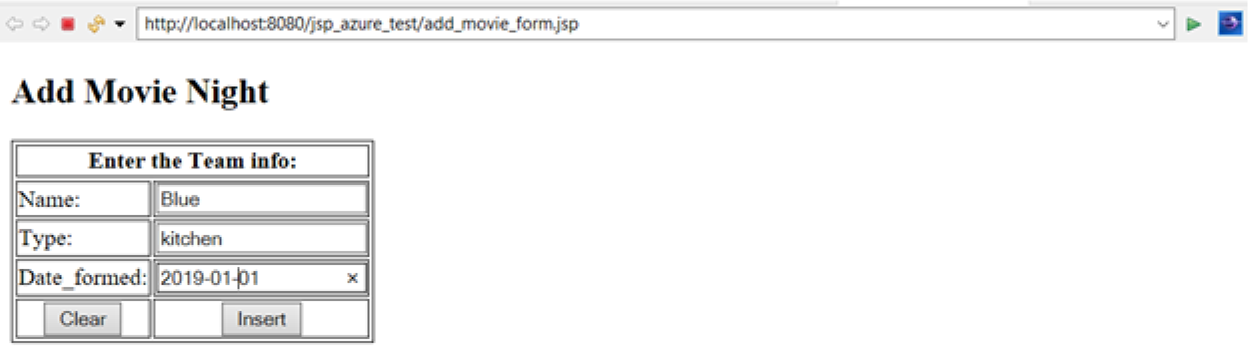
- Get the initial data:



http://localhost:8080/jsp_azure_test/get_all_movies.jsp

Team Name	Type	Date Formed
Red	Clean	2019-12-31 20:00:00.0
Yellow	Medical	2020-01-03 19:00:00.0

- Insert page:

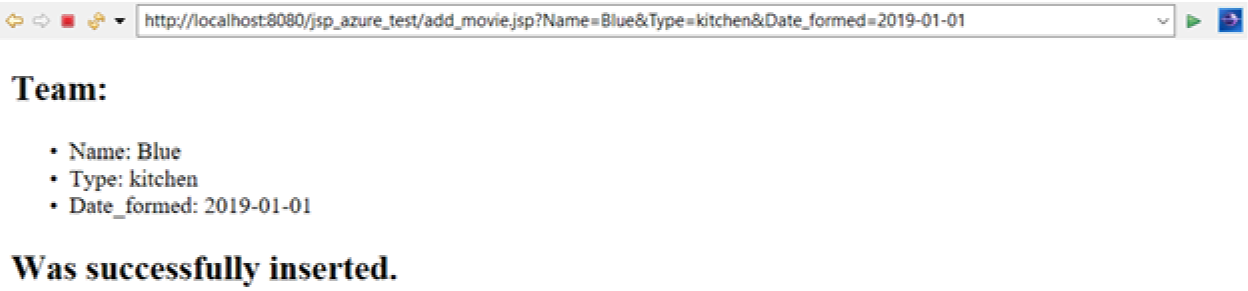


http://localhost:8080/jsp_azure_test/add_movie_form.jsp

Add Movie Night

Enter the Team info:	
Name:	Blue
Type:	kitchen
Date_formed:	2019-01-01

- Insert Success page



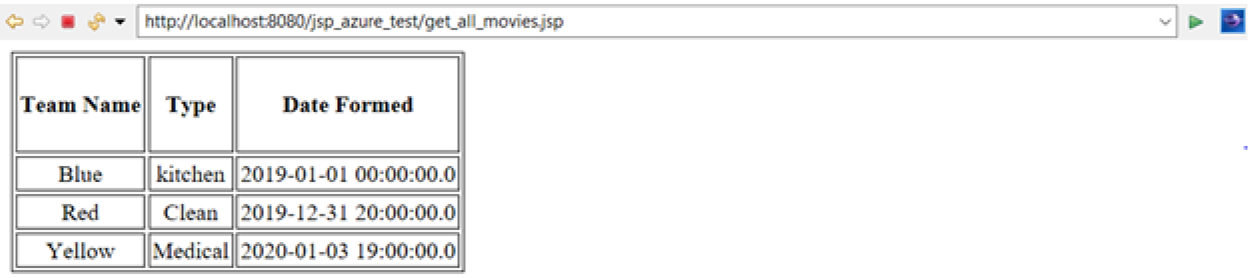
http://localhost:8080/jsp_azure_test/add_movie.jsp?Name=Blue&Type=kitchen&Date_formed=2019-01-01

Team:

- Name: Blue
- Type: kitchen
- Date_formed: 2019-01-01

Was successfully inserted.

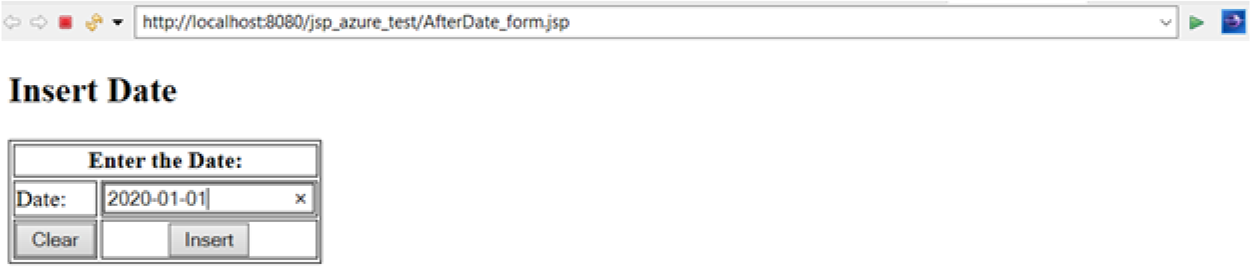
- Table after insert:



http://localhost:8080/jsp_azure_test/get_all_movies.jsp

Team Name	Type	Date Formed
Blue	kitchen	2019-01-01 00:00:00.0
Red	Clean	2019-12-31 20:00:00.0
Yellow	Medical	2020-01-03 19:00:00.0

- Date form:



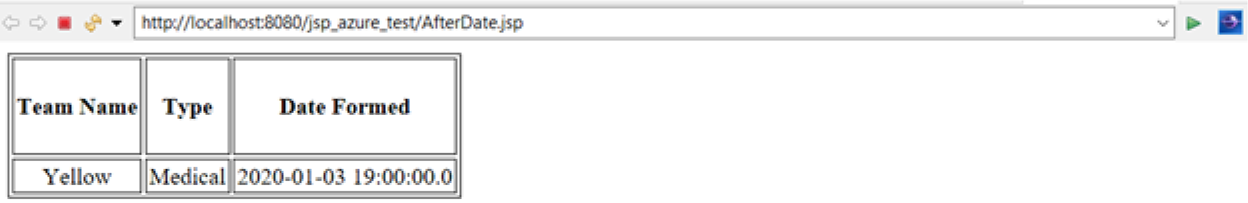
http://localhost:8080/jsp_azure_test/AfterDate_form.jsp

Insert Date

Enter the Date:

Date: 2020-01-01

- Table for date after entered date



http://localhost:8080/jsp_azure_test/AfterDate.jsp

Team Name	Type	Date Formed
Yellow	Medical	2020-01-03 19:00:00.0