

**Course Name and Number: Database Management Systems — DSA  
4513-001**

**Section Number: 001**

**Semester and Year: Fall 2024**

**Instructor Name: Dr. Le Gruenwald**

**Author Name: Roshini Talluru**

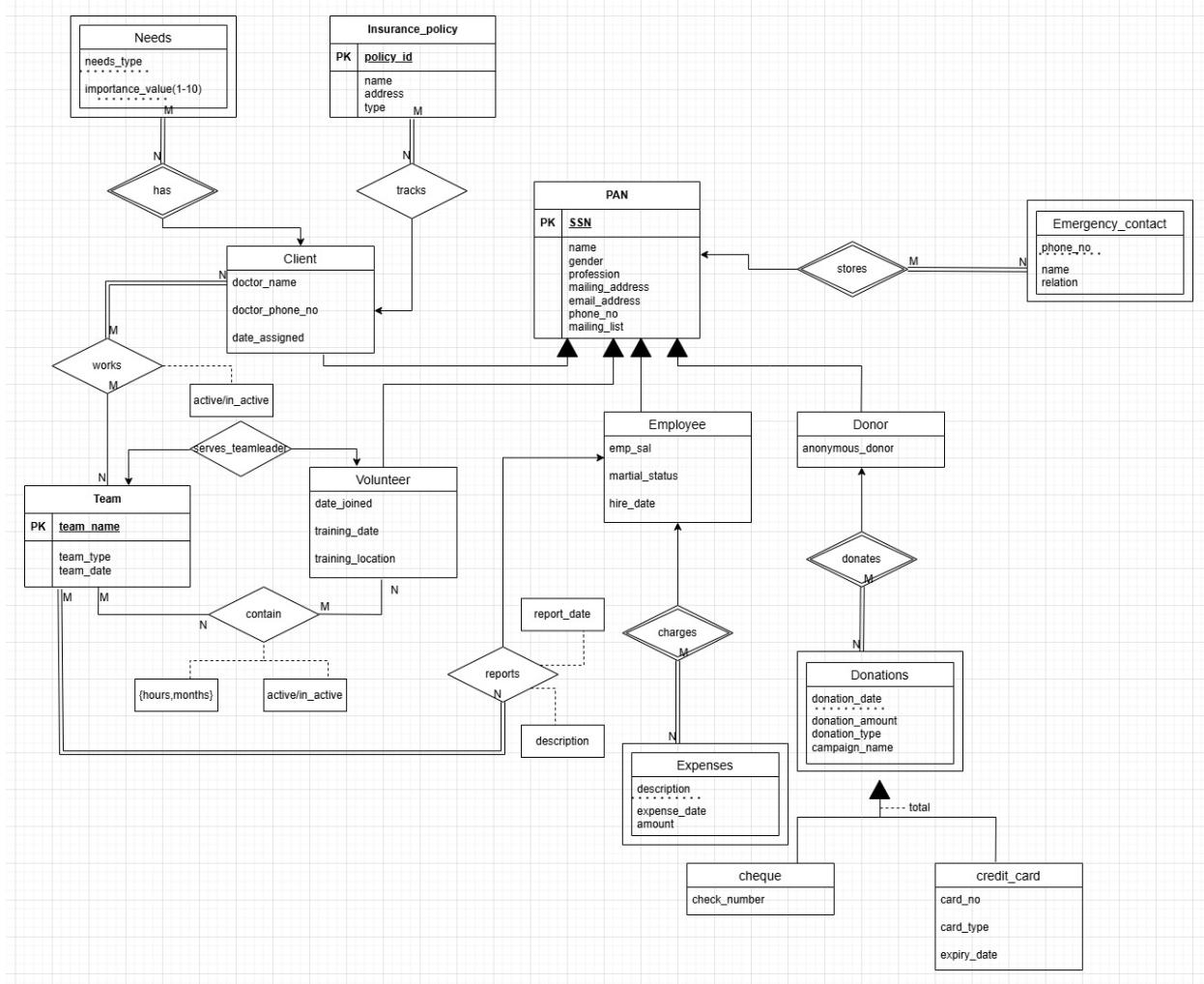
**Id: 113582022**

**Email: [roshini.talluru-1@ou.edu](mailto:roshini.talluru-1@ou.edu)**

**Title: A PATIENT ASSISTANT NETWORK DATABASE SYSTEM**

<b>Tasks Performed</b>	<b>Page Number</b>
Task 1. ER Diagram	3
Task 2. Relational Database Schemas	4
Task 3.	
3.1. Discussion of storage structures for tables	5-9
3.2. Discussion of storage structures for tables (Azure SQL Database)	10-11
Task 4. SQL statements and screenshots showing the creation of tables in Azure SQL Database	12-21
Task 5.	22-28
5.1 SQL statements ( <b>and Transact SQL stored procedures, if any</b> )	22-37
Implementing all queries (1-15 and error checking)	
5.2 The Java source program and screenshots showing its successful compilation	38-48
Task 6. Java program Execution	49-90
6.1. Screenshots showing the testing of query 1	49-51
6.2. Screenshots showing the testing of query 2	52-58
6.3. Screenshots showing the testing of query 3	58-63
6.4. Screenshots showing the testing of query 4	63-67
6.5. Screenshots showing the testing of query 5	67-73
6.6. Screenshots showing the testing of query 6	73-78
6.7. Screenshots showing the testing of query 7	78-85
6.8. Screenshots showing the testing of query 8	85-86
6.9. Screenshots showing the testing of query 9	86
6.10. Screenshots showing the testing of query 10	86-87
6.11. Screenshots showing the testing of query 11	87-88
6.12. Screenshots showing the testing of query 12	88-89
6.13. Screenshots showing the testing of query 13	89
6.14. Screenshots showing the testing of query 14	89-90
6.15. Screenshots showing the testing of query 15	91-92
6.16. Screenshots showing the testing of the import and export options	92-84
6.17. Screenshots showing the testing of three types of errors	95
6.18. Screenshots showing the testing of the quit option	95

**Task 1. (75 points): Design an ER diagram to represent the PAN database defined in Section I.**



**Task 2. (30 Points): Convert the ER diagram in Task 1 to a Relational Database (i.e. a set of relational schemas).**

PAN(SSN, name, gender, profession, mailing\_address, email\_address, phone\_no, mailing\_list)

Emergency\_contact(SSN, phone\_no, name, relation)

Client(SSN, doctor\_name, doctor\_phone\_no, date\_assigned)

Needs(Client\_SSN, needs\_type, importance\_value(1-10))

Insurance\_policy(policy\_id, Client\_SSN, name, address, type)

Voluteer(SSN, date\_joined, training\_date, training\_location)

Teams(team\_name, team\_type, team\_date)

Contain(team\_name, Voluteer\_SSN, active/inactive)

Hour/month(team\_name, Voluteer\_SSN, hours, months)

Serves\_Team\_leader(team\_name, Voluteer\_SSN)

Works(team\_name, Client\_SSN, active/in\_active)

Employee(SSN, emp\_sal, marital\_status, hire\_date)

Reports(team\_name, Employee\_SSN, report\_date, description)

Expenses(Employee\_SSN, description, expense\_date, amount)

Donor(SSN, anonymous\_donor)

Cheque(Donor\_SSN, donation\_date, donation\_amount, donation\_type, campaign\_name, cheque\_number)

Credit\_card(Donor\_SSN, donation\_date, donation\_amount, donation\_type, campaign\_name, card\_no, card\_type, expiry\_date)

**Task 3. (40 points):**

**3.1. Discuss choices of appropriate storage structures for each relational table assuming that all types of storage structures discussed in class (Lecture Topic 4) are available. For each table, identify the queries (from the list of the given queries) that access the table, the type of each of those queries (insertion, deletion, random search, or range search), the search keys (if any) involved in each of those queries, the frequency of each of those queries, your choice of the file organization for the table, and your detailed justifications. Use the following format to fill out your answers:**

Let's assume 1- insertion, 2-deletion, 3-random search and 4-range search.

Table Name	Query # and Type	Search_Key	Query Frequency	Selected File Organization	Justification
PAN	12 and 3 2 and 1 3 and 1 5 and 1 7 and 1	SSN	1/week 1/week 2/month 1/year 1/day	Dynamic Hashing with search key SSN	Dynamic hashing provides fast access for unique identifier-based lookups, which is efficient for frequent insertions and retrievals.
Emergency Contact	12 and 3	SSN(PAN)	1/week	Dynamic Hashing with search key SSN	Supports efficient random searches, allowing for quick retrieval of contact information based on a unique identifier.
Client	2 and 1 8 and 3 10 and 3	SSN(PAN)	1/week 1/week 4/year 4/year	Dynamic Hashing with search key SSN	Dynamic hashing provides fast access for unique identifier-based

	15 and 2				lookups, which is efficient for frequent insertions and retrievals.
Needs	15 and 4	Importance_value(1 -10)	4/year	Sequential Indexing on importance_value	Sequential indexing is ideal for range-based with minimal updates.
Insurance_policy	15 and 3	type	4/year	Sequential on secondary index type	Primary index on policy_id allows direct lookups. Secondary index on type supports efficient categorization or filtering by policy type for future queries.
Volunteer	3 and 1 10 and 3	SSN(PAN)	2/month 4/year	Dynamic hashing on SSN	Dynamic hashing supports efficient random access to unique SSNs
Teams	1 and 1 11 and 4	Team_date	1/month 1/month	Sequential indexing on secondary_index team_date	Sequential indexing is ideal for range-based with minimal updates
contain	3 and 1 4 and 1 10 and 3	Team_name	2/month 30/month 4/year	B+ Tree on team_name	Balances frequent insertions with efficient ordered access, making it suitable for retrievals that

					need to find associations between volunteers and teams.
Hour/month	4 and 1	No search key	30/month	Heap File organization	Handles high insertion rates efficiently without the need for ordering, making it ideal for logging hours without complex retrieval requirements.
Serves_team_leade r	No query and 1	No search key	infrequent	Heap File organization	Handles high insertion rates efficiently without the need for ordering, making it ideal for logging hours without complex retrieval requirements.
works	2 and 1 10 and 3	Client_SSN	1/week 4/year	Dynamic Hashing	Dynamic hashing allows fast insertions and efficient retrieval of team-client relationships by client identifier.

Employee	5 and 1 6 and 3 9 and 4 13 and 3 14 and 3	SSN(PAN)	1/year 1/day 1/month 1/week 1/year	B+ Tree on SSN(PAN)	B+ Tree supports efficient ordered retrievals and moderate updates, making it ideal for managing employee records with both insertion and retrieval needs.
Reports	5 and 1 14 and 3	Team_name	1/year 1/year	Dynamic hashing on team_name	Dynamic hashing provides fast access for unique identifier-based lookups, which is efficient for frequent insertions and retrievals.
expenses	6 and 1 9 and 4	Expense_date	1/day 1/month	B+Tree on expense_date	B+ Tree supports efficient range and exact queries by expense_date , making it ideal for date-based retrievals and moderate insertions.
Donor	7 and 1 13 and 3	SSN(PAN)	1/day 1/week	Dynamic Hashing on SSN(PAN)	efficient for frequent insertions and fast access to

					donor information by unique identifier.
Cheque	7 and 1 13 and 3	Donor_SSN	1/day 1/week	Dynamic Hashing on Donor_SSN	Supports high insertion rates and efficient lookups for donation details related to checks, based on donor identifier.
Credit_card	7 and 1 13 and 3	Donor_SSN	1/day 1/week	Dynamic Hashing on Donor_SSN	Similar to cheque entries, Dynamic Hashing supports frequent insertions and fast retrieval of donation details for credit card entries.

**3.2. Discuss choices of storage structures for each relational table when implementing it in Azure SQL Database (if different from the previous choices specified in 3.1). Part of this task is for you to find and study the relevant documentation on your own.**

There are indeed few file organization changes which is list below in the table with justification.

Table Name	Search_Key	Original File Organization	Revised file Organization	Justification
PAN	SSN	Dynamic Hashing with search key SSN	B + Tree on SSN	B+ Tree provides efficient lookups on unique identifiers and supports Azure SQL indexing limitations.frequent insertions and retrievals.
Emergency Contact	SSN(PAN)	Dynamic Hashing with search key SSN	B+ Tree on SSN	B+ Tree offers efficient lookups by unique identifiers; clustered index on SSN aligns with Azure SQL capabilities.
Client	SSN(PAN)	Dynamic Hashing with search key SSN	B+Tree on SSN	B+ Tree enables efficient insertions and lookups on SSN; clustered index supports primary access by client ID.
Needs	Importance_value(1-10)	Sequential Indexing on importance_valu	No change	N/A
Insurance_policy	type	Sequential indexing	No change	N/A
Volunteer	SSN(PAN)	Dynamic hashing on SSN	B+Tree	B+ Tree with clustered index supports unique identifiers and frequent lookups, improving retrieval efficiency.
Teams	Team_date	Sequential indexing	No Change	N/A
contain	Team_name	B+ Tree on team_name	No Change	N/A
Hour/month	No search key	Heap File organization	No change	N/A

Serves_team_leader	No search key	Heap File organization	No change	N/A
works	Client_SSN	Dynamic Hashing	B+Tree	B+ Tree with clustered index enables fast insertions and retrieval of team-client relationships by Client_SSN.
Employee	SSN(PAN)	B+ Tree on SSN(PAN)	No change	N/A
Reports	Team_name	Dynamic hashing on team_name	B+ Tree on team_name	B+ Tree with clustered index on team_name allows efficient retrievals of team-based reports with Azure SQL support.
expenses	Expense_date	B+Tree on expense_date	No change	N/A
Donor	SSN(PAN)	Dynamic Hashing on SSN(PAN)	No change	N/A
Cheque	Donor_SSN	Dynamic Hashing on Donor_SSN	No change	N/A.
Credit_card	Donor_SSN	Dynamic Hashing on Donor_SSN	No Change	N/A

Apart from this, I thought of using default clustered index on the primary key. Since, we can only create one clustered index per table, we can create non-clustered instead for the following tables:

Insurance\_policy table:

- Default clustered index on the primary key ‘policy\_id’.
- Non-clustered index on ‘type’.

Teams Table:

- Default clustered index on ‘team\_name’.
- Non-clustered index on ‘team\_date’

Expenses Table:

- Default clustered index on ‘employee\_ssn’ and ‘description’.
- Non-clustered index on ‘expense\_date’.

**Task 4. (25 points): Construct SQL statements to create tables and implement them on Azure SQL Database. All CREATE statements must include appropriate constraints as defined in Task 2. For each table, you must include SQL statements that create the same storage structure as the one you selected for Azure SQL Database implementation in Task 3.2 (e.g., if you have decided that a table X must have an index on attribute Y, then you must include an SQL statement to create an index on attribute Y for table X).**

--Task 4

```
-- Drop tables in the correct order to handle dependencies
DROP TABLE IF EXISTS Credit_card;
DROP TABLE IF EXISTS Cheque;
DROP TABLE IF EXISTS Donor;
DROP TABLE IF EXISTS Expenses;
DROP TABLE IF EXISTS Reports;
DROP TABLE IF EXISTS Employee;
DROP TABLE IF EXISTS Works;
DROP TABLE IF EXISTS Serves_Team_leader;
DROP TABLE IF EXISTS Hour_month;
DROP TABLE IF EXISTS Contain;
DROP TABLE IF EXISTS Teams;
DROP TABLE IF EXISTS Volunteer;
DROP TABLE IF EXISTS Insurance_policy;
DROP TABLE IF EXISTS Needs;
DROP TABLE IF EXISTS Client;
DROP TABLE IF EXISTS Emergency_contact;
DROP TABLE IF EXISTS PAN;
```

#### 1. PAN Table

```
23    ---PAN Table
24    CREATE TABLE PAN (
25        SSN VARCHAR(20) PRIMARY KEY,
26        name VARCHAR(50),
27        gender CHAR(1) CHECK (gender IN ('M', 'F')),
28        profession VARCHAR(100),
29        mailing_address VARCHAR(200),
30        email_address VARCHAR(100),
31        phone_no VARCHAR(15),
32        mailing_list BIT
33    );
34
```

## Messages

3:13:32 AM     Started executing query at Line 23  
Commands completed successfully.  
Total execution time: 00:00:00.077

## 2. Emergency\_Contact Table

```
35    --Emergency Contact
36    CREATE TABLE Emergency_contact (
37        SSN VARCHAR(20),
38        phone_no VARCHAR(15),
39        name VARCHAR(50),
40        relation VARCHAR(50),
41        PRIMARY KEY (SSN, phone_no),
42        FOREIGN KEY (SSN) REFERENCES PAN(SSN)
43    );
44
```

## Messages

3:14:15 AM     Started executing query at Line 35  
Commands completed successfully.  
Total execution time: 00:00:00.068

## 3. Client table

```
45  --client
46  CREATE TABLE Client (
47      SSN VARCHAR(20) PRIMARY KEY,
48      doctor_name VARCHAR(50),
49      doctor_phone_no VARCHAR(15),
50      date_assigned DATE,
51      FOREIGN KEY (SSN) REFERENCES PAN(SSN)
52  );
53
```

## Messages

3:14:47 AM     Started executing query at Line 45  
Commands completed successfully.  
Total execution time: 00:00:00.075

## 4. Needs Table

```
54  --Needs
55  CREATE TABLE Needs (
56      Client_SSN VARCHAR(20),
57      needs_type VARCHAR(50),
58      importance_value TINYINT CHECK (importance_value BETWEEN 1 AND 10),
59      PRIMARY KEY (Client_SSN, needs_type, importance_value),
60      FOREIGN KEY (Client_SSN) REFERENCES Client(SSN)
61  );
62
```

## Messages

3:15:15 AM     Started executing query at Line 54  
Commands completed successfully.  
Total execution time: 00:00:00.059

## 5. Insurance\_policy Table

```
63    --Insurance_policy
64    CREATE TABLE Insurance_policy (
65        policy_id INT PRIMARY KEY, -- clustered index on policy_id
66        Client_SSN VARCHAR(20),
67        name VARCHAR(50),
68        address VARCHAR(200),
69        type VARCHAR(50) CHECK (type IN ('life', 'health', 'home','auto')),
70        FOREIGN KEY (Client_SSN) REFERENCES Client(SSN)
71    );
72
```

## Messages

3:15:54 AM     [Started executing query at Line 63](#)  
Commands completed successfully.  
Total execution time: 00:00:00.081

### Create non-clustered index on 'type':

```
73    CREATE NONCLUSTERED INDEX idx_type ON Insurance_policy(type); -- Non-clustered index on type
74
```

## Messages

3:16:52 AM     [Started executing query at Line 73](#)  
Commands completed successfully.  
Total execution time: 00:00:00.064

## 6. Volunteer

```
75    --Volunteer
76    CREATE TABLE Volunteer (
77        SSN VARCHAR(20) PRIMARY KEY,
78        date_joined DATE,
79        training_date DATE,
80        training_location VARCHAR(100),
81        FOREIGN KEY (SSN) REFERENCES PAN(SSN)
82    );
83
```

## Messages

```
3:17:51 AM      Started executing query at Line 75
                  Commands completed successfully.
                  Total execution time: 00:00:00.317
```

## 7. Team table

```
84    --Team
85    CREATE TABLE Teams (
86        team_name VARCHAR(50) PRIMARY KEY, -- Clustered index on team_name
87        team_type VARCHAR(50),
88        team_date DATE
89    );
90
```

## Messages

```
3:18:36 AM      Started executing query at Line 84
                  Commands completed successfully.
                  Total execution time: 00:00:00.069
```

### Create non-clustered index on 'team\_date':

```
91      CREATE NONCLUSTERED INDEX idx_team_date ON Teams(team_date); -- Non-clustered index on team_date
92
```

## Messages

```
3:19:21 AM      Started executing query at Line 91
                  Commands completed successfully.
                  Total execution time: 00:00:00.074
```

## 8. Contain Table

```
93 --Contain
94 CREATE TABLE Contain (
95     team_name VARCHAR(50),
96     Volunteer_SSN VARCHAR(20),
97     active_inactive VARCHAR(10) Default 1, -- 1for active, 0 for inactive
98     PRIMARY KEY (team_name, Volunteer_SSN),
99     FOREIGN KEY (team_name) REFERENCES Teams(team_name),
100    FOREIGN KEY (Volunteer_SSN) REFERENCES Volunteer(SSN)
101 );
102
```

## Messages

3:21:14 AM      Started executing query at Line 93  
Commands completed successfully.  
Total execution time: 00:00:00.088

## **9. Hour/Month**

## Messages

3:22:17 AM      Started executing query at Line 103  
Commands completed successfully.  
Total execution time: 00:00:00.094

## **10. Team Leader**

```
115  -- Team Leader
116  CREATE TABLE Serves_Team_leader (
117      team_name VARCHAR(50),
118      Volunteer_SSN VARCHAR(10),
119      PRIMARY KEY (team_name),
120      FOREIGN KEY (team_name) REFERENCES Teams(team_name)
121  );
122
```

## Messages

3:23:12 AM      Started executing query at Line 115  
Commands completed successfully.  
Total execution time: 00:00:00.073

## 11. Works Table

```
123  --Works
124  CREATE TABLE Works (
125      team_name VARCHAR(50),
126      Client_SSN VARCHAR(20),
127      active_inactive VARCHAR(10) DEFAULT 1,-- 1 for active, 0 -inactive
128      PRIMARY KEY (team_name, Client_SSN),
129      FOREIGN KEY (team_name) REFERENCES Teams(team_name),
130      FOREIGN KEY (Client_SSN) REFERENCES Client(SSN)
131  );
132  Employees
```

## Messages

3:24:26 AM      Started executing query at Line 124  
Commands completed successfully.  
Total execution time: 00:00:00.071

## 12. Employee Table

```
132    --Employee
133    CREATE TABLE Employee (
134        SSN VARCHAR(20) PRIMARY KEY,
135        emp_sal DECIMAL(10, 2),
136        marital_status VARCHAR(10),
137        hire_date DATE,
138        FOREIGN KEY (SSN) REFERENCES PAN(SSN)
139    );
140
```

## Messages

3:24:57 AM     Started executing query at Line 132  
Commands completed successfully.  
Total execution time: 00:00:00.069

### 13. Reports Table

```
141    --Reports
142    CREATE TABLE Reports (
143        team_name VARCHAR(50) PRIMARY KEY,
144        Employee_SSN VARCHAR(20),
145        report_date DATE,
146        description VARCHAR(255),
147        FOREIGN KEY (team_name) REFERENCES Teams(team_name),
148        FOREIGN KEY (Employee_SSN) REFERENCES Employee(SSN)
149    );
150
```

## Messages

3:25:23 AM     Started executing query at Line 141  
Commands completed successfully.  
Total execution time: 00:00:00.065

### 14. Expenses Table

```
151    --Expenses
152    CREATE TABLE Expenses (
153        Employee_SSN VARCHAR(20),
154        description VARCHAR(255),
155        expense_date DATE,
156        amount DECIMAL(10, 2),
157        PRIMARY KEY (Employee_SSN, description),
158        FOREIGN KEY (Employee_SSN) REFERENCES Employee(SSN)
159    );
```

## Messages

3:25:48 AM     Started executing query at Line 151  
Commands completed successfully.  
Total execution time: 00:00:00.061

### Create non-clustered index on 'expense\_date':

```
161    CREATE NONCLUSTERED INDEX idx_expense_date ON Expenses(expense_date); -- Non-clustered index on expense_date
162
```

## Messages

3:26:29 AM     Started executing query at Line 161  
Commands completed successfully.  
Total execution time: 00:00:00.064

## 15. Donor Table

```
163    --Donor
164    CREATE TABLE Donor (
165        SSN VARCHAR(20) PRIMARY KEY,
166        anonymous_donor BIT,
167        FOREIGN KEY (SSN) REFERENCES PAN(SSN)
168    );
169
```

## Messages

3:27:28 AM     Started executing query at Line 163  
Commands completed successfully.  
Total execution time: 00:00:00.064

## 16. Cheque Table

```
170    --Cheque
171    CREATE TABLE Cheque (
172        Donor_SSN VARCHAR(20),
173        donation_date DATE,
174        donation_amount DECIMAL(10, 2),
175        donation_type VARCHAR(50),
176        campaign_name VARCHAR(100),
177        cheque_number INT,
178        PRIMARY KEY (Donor_SSN, donation_date, cheque_number),
179        FOREIGN KEY (Donor_SSN) REFERENCES Donor(SSN)
180    );
181
```

### Messages

3:28:17 AM     Started executing query at Line 170  
Commands completed successfully.  
Total execution time: 00:00:00.070

## 17. Credit Card Table

```
182    --Credit Card Table
183    CREATE TABLE Credit_card (
184        Donor_SSN VARCHAR(20),
185        donation_date DATE,
186        donation_amount DECIMAL(10, 2),
187        donation_type VARCHAR(50),
188        campaign_name VARCHAR(100),
189        card_no VARCHAR(20),
190        card_type VARCHAR(20),
191        expiry_date DATE,
192        PRIMARY KEY (Donor_SSN, donation_date, card_no),
193        FOREIGN KEY (Donor_SSN) REFERENCES Donor(SSN)
194    );
195
```

### Messages

3:28:53 AM     Started executing query at Line 182  
Commands completed successfully.  
Total execution time: 00:00:00.073

**Task 5. (Task 5 and Task 6 together = 130 points):** Write SQL statements for all queries (1-15) defined in part I. Write a Java application program that uses JDBC and Azure SQL Database to implement all SQL queries (options 1-15), two additional queries for import and export (options 16 17), and the “Quit” option (option 18) as specified in the menu given below. You are free to pick any file format you wish to use for file import and export options. The program will stop execution only when the user chooses the “Quit” option; otherwise, all options must be available for the user to choose at all times. Your program must be commented properly.

### 5.1. SQL statements and Transact SQL stored procedures

#### 1. Enter a new team into the database (1/month).

```
--Query 1
DROP PROCEDURE IF EXISTS AddNewTeam;
GO
CREATE PROCEDURE AddNewTeam
    @team_name VARCHAR(50),
    @team_type VARCHAR(50),
    @team_date DATE
AS
BEGIN
    -- Check if any team exists with the same team_name
    IF EXISTS (SELECT 1 FROM Teams
    |   |   WHERE team_name = @team_name
    )
    -- Insert team
    INSERT INTO Teams (team_name, team_type, team_date)
    VALUES (@team_name, @team_type, @team_date);

    PRINT 'Team successfully added.';
END;
SELECT * FROM Teams;-- for testing
GO
```

---

## Messages

3:38:15 AM	<u>Started executing query at Line 199</u> Commands completed successfully.
3:38:15 AM	<u>Started executing query at Line 201</u> Commands completed successfully. Total execution time: 00:00:00.143

#### 2. Enter a new client into the database and associate him or her with one or more teams (1/week).

```

225  --Query 2
226  DROP PROCEDURE IF EXISTS AddNewClientWithTeams;
227  GO
228  CREATE PROCEDURE AddNewClientWithTeams
229      @SSN VARCHAR(20),
230      @name VARCHAR(50),
231      @gender CHAR(1),
232      @profession VARCHAR(100),
233      @mailing_address VARCHAR(200),
234      @email_address VARCHAR(100),
235      @phone_no VARCHAR(15),
236      @mailing_list BIT,
237      @doctor_name VARCHAR(50),
238      @doctor_phone_no VARCHAR(15),
239      @date_assigned DATE,
240      @team_names NVARCHAR(MAX) ,-- Comma-separated list of team names
241      @active_inactive BIT
242  AS
243  BEGIN
244      -- PAN Table insertion
245      BEGIN TRY
246          INSERT INTO PAN (SSN, name, gender, profession, mailing_address, email_address, phone_no, mailing_list)
247              VALUES (@SSN, @name, @gender, @profession, @mailing_address, @email_address, @phone_no, @mailing_list);
248      END TRY
249      BEGIN CATCH
250          PRINT 'Error inserting into PAN table';
251          RETURN;
252      END CATCH;
253
254      -- Client Table Insertion
255      BEGIN TRY
256          INSERT INTO Client (SSN, doctor_name, doctor_phone_no, date_assigned)
257              VALUES (@SSN, @doctor_name, @doctor_phone_no, @date_assigned);
258      END TRY
259      BEGIN CATCH
260          PRINT 'Error inserting into Client table';
261          RETURN;
262      END CATCH;
263      -- Insert each team association into the Works table
264      DECLARE @team_name VARCHAR(50);
265      DECLARE @pos INT;
266
267      SET @team_names = @team_names + ','; -- Add comma for easier parsing
268      WHILE CHARINDEX(',', @team_names) > 0
269      BEGIN
270          SET @pos = CHARINDEX(',', @team_names);
271          SET @team_name = LTRIM(RTRIM(SUBSTRING(@team_names, 1, @pos - 1)));
272          SET @team_names = SUBSTRING(@team_names, @pos + 1, LEN(@team_names));
273
274          -- Verify if the team exists before associating it with the client
275          IF EXISTS (SELECT 1 FROM Teams WHERE team_name = @team_name)
276          BEGIN
277              BEGIN TRY
278                  INSERT INTO Works (team_name, Client_SSN, active_inactive)
279                      VALUES (@team_name, @SSN, @active_inactive);
280              END TRY
281              BEGIN CATCH
282                  PRINT 'Error inserting into Works table for team: ' + @team_name;

```

```
282      |      END CATCH;
283      |      END
284      |      ELSE
285      |      BEGIN
286      |          PRINT 'Team not found: ' + @team_name;
287      |      END
288      END;
289 END;
290 --Testing
291 SELECT * from Client;
292 Select * from Works;
293 Select *from PAN;
294 Select *from Teams;
295 GO
296
```

204      1151

## Messages

3:42:42 AM	<u><a href="#">Started executing query at Line 226</a></u> Commands completed successfully.
3:42:42 AM	<u><a href="#">Started executing query at Line 228</a></u> Commands completed successfully. Total execution time: 00:00:00.144

- 3. Enter a new volunteer into the database and associate him or her with one or more teams (2/month).**

```

297  --Query 3
298  DROP PROCEDURE IF EXISTS AddNewVolunteerWithTeams;
299  GO
300  CREATE PROCEDURE AddNewVolunteerWithTeams
301      @SSN VARCHAR(20),
302      @name VARCHAR(50),
303      @gender CHAR(1),
304      @profession VARCHAR(100),
305      @mailing_address VARCHAR(200),
306      @email_address VARCHAR(100),
307      @phone_no VARCHAR(15),
308      @mailing_list BIT,
309      @date_joined DATE,
310      @training_date DATE,
311      @training_location VARCHAR(100),
312      @team_names NVARCHAR(MAX),
313      @active_inactive BIT -- Comma-separated list of team names
314  AS
315  BEGIN
316      -- Insertion PAN table
317      BEGIN TRY
318          INSERT INTO PAN (SSN, name, gender, profession, mailing_address, email_address, phone_no, mailing_list)
319          VALUES (@SSN, @name, @gender, @profession, @mailing_address, @email_address, @phone_no, @mailing_list);
320      END TRY
321      BEGIN CATCH
322          PRINT 'Error inserting into PAN table';
323          -
324      END CATCH;
325
326      -- Insertion Volunteer table
327      BEGIN TRY
328          INSERT INTO Volunteer (SSN, date_joined, training_date, training_location)
329          VALUES (@SSN, @date_joined, @training_date, @training_location);
330      END TRY
331      BEGIN CATCH
332          PRINT 'Error inserting into Volunteer table';
333          RETURN;
334      END CATCH;
335
336      -- Insert each team association into the Contain table
337      DECLARE @team_name VARCHAR(50);
338      DECLARE @pos INT;
339
340      SET @team_names = @team_names + ','; -- Add comma for easier parsing
341      WHILE CHARINDEX(',', @team_names) > 0
342      BEGIN
343          SET @pos = CHARINDEX(',', @team_names);
344          SET @team_name = LTRIM(RTRIM(SUBSTRING(@team_names, 1, @pos - 1)));
345          SET @team_names = SUBSTRING(@team_names, @pos + 1, LEN(@team_names));
346
347          -- Verify if the team exists before associating it with the volunteer
348          IF EXISTS (SELECT 1 FROM Teams WHERE team_name = @team_name)
349          BEGIN
350              BEGIN TRY

```

```

354     BEGIN CATCH
355         PRINT 'Error inserting into Contain table for team: ' + @team_name;
356     END CATCH;
357     END
358     ELSE
359     BEGIN
360         PRINT 'Team not found: ' + @team_name;
361     END
362 END;
363 END;
364 SELECT * from Volunteer;
365 Select * from Contain;
366 Select *from PAN;
367 Select *from Teams;
368 GO
369

```

## Messages

3:45:43 AM	<u>Started executing query at Line 299</u>
	Commands completed successfully.
3:45:43 AM	<u>Started executing query at Line 301</u>
	Commands completed successfully.
	Total execution time: 00:00:00.171

## 4. Enter the number of hours a volunteer worked this month for a particular team (30/month).

```

370 --Query 4
371 DROP PROCEDURE IF EXISTS EnterVolunteerHours;
372 GO
373 CREATE PROCEDURE EnterVolunteerHours
374     (@team_name VARCHAR(50),
375      @Volunteer_SSN VARCHAR(20),
376      @hours INT,
377      @month VARCHAR(3) -- e.g., 'Jan', 'Feb', etc.
378 AS
379     -- Check if the volunteer is associated with the specified team
380     IF NOT EXISTS (SELECT 1
381                     FROM Contain
382                     WHERE team_name = @team_name
383                     AND Volunteer_SSN = @Volunteer_SSN)
384     BEGIN
385         RAISERROR('Volunteer with SSN %s is not associated with team %s.', 16, 1, @Volunteer_SSN, @team_name);
386         RETURN;
387     END
388 BEGIN
389     -- Check if a record with the same hours already exists for this volunteer and team
390     IF EXISTS (SELECT 1
391                     FROM Hour_month
392                     WHERE team_name = @team_name
393                     AND Volunteer_SSN = @Volunteer_SSN
394                     AND hours = @hours)
395     BEGIN
396         RAISERROR('This volunteer already logged %d hours for the team %s in a previous month. Please enter a different number of hours.', 16, 1, @hours, @team_name);
397         RETURN;
398     END
399
400

```

```

-- Insert the hours worked for the volunteer in the specified month if all checks pass
401 BEGIN TRY
402     INSERT INTO Hour_month (team_name, Volunteer_SSN, hours, months)
403     VALUES (@team_name, @Volunteer_SSN, @hours, @month);
404
405     PRINT 'Hours successfully added for the volunteer.';
406 END TRY
407 BEGIN CATCH
408     RAISERROR('Error inserting hours for Volunteer SSN: %s in team: %s', 16, 1, @Volunteer_SSN, @team_name);
409 END CATCH;
410
411 END;
412 GO
413 SELECT * FROM Contain;
414 select *from Volunteer;
415 select * from Hour_month;
416 GO

```

3:47:59 AM      Started executing query at Line 371  
                   Commands completed successfully.

3:47:59 AM      Started executing query at Line 373  
                   Commands completed successfully.

3:47:59 AM      Started executing query at Line 414  
                   (0 rows affected)  
                   (0 rows affected)  
                   (0 rows affected)

Total execution time: 00:00:00.222

## 5. Enter a new employee into the database and associate him or her with one or more teams

(1/year).

```

418 --Query 5
419 DROP PROCEDURE IF EXISTS AddNewEmployeeWithTeams;
420 GO
421 CREATE PROCEDURE AddNewEmployeeWithTeams
422     @SSN VARCHAR(20),
423     @name VARCHAR(50),
424     @gender CHAR(1),
425     @profession VARCHAR(100),
426     @mailing_address VARCHAR(200),
427     @email_address VARCHAR(100),
428     @phone_no VARCHAR(15),
429     @mailing_list BIT,
430     @emp_sal DECIMAL(10, 2),
431     @marital_status VARCHAR(10),
432     @hire_date DATE,
433     @team_names NVARCHAR(MAX),
434     @report_date DATE,
435     @description VARCHAR(255)
436 AS
437 BEGIN
438     -- PAN table
439     BEGIN TRY
440         IF NOT EXISTS (SELECT 1 FROM PAN WHERE SSN = @SSN)
441             BEGIN
442                 INSERT INTO PAN (SSN, name, gender, profession, mailing_address, email_address, phone_no, mailing_list)
443                     VALUES (@SSN, @name, @gender, @profession, @mailing_address, @email_address, @phone_no, @mailing_list);
444             END
445     END TRY
446     BEGIN CATCH

```

```

446    BEGIN CATCH
447        PRINT 'Error inserting into PAN table';
448        RETURN;
449    END CATCH;
450
451    -- Employee table
452    BEGIN TRY
453        IF NOT EXISTS (SELECT 1 FROM Employee WHERE SSN = @SSN)
454        BEGIN
455            INSERT INTO Employee (SSN, emp_sal, marital_status, hire_date)
456            VALUES (@SSN, @emp_sal, @marital_status, @hire_date);
457        END
458    END TRY
459    BEGIN CATCH
460        PRINT 'Error inserting into Employee table';
461        RETURN;
462    END CATCH;
463
464    -- Insert each team association into the Reports table
465    DECLARE @team_name VARCHAR(50);
466    DECLARE @pos INT;
467
468    SET @team_names = @team_names + ','; -- Add comma for easier parsing
469    WHILE CHARINDEX(',', @team_names) > 0
470    BEGIN
471        SET @pos = CHARINDEX(',', @team_names);
472        SET @team_name = LTRIM(RTRIM(SUBSTRING(@team_names, 1, @pos - 1)));
473        SET @team_names = SUBSTRING(@team_names, @pos + 1, LEN(@team_names));
474
475        -- Verify if the team exists before associating it with the employee
476        IF EXISTS (SELECT 1 FROM Teams WHERE team_name = @team_name)
477        BEGIN
478            BEGIN TRY
479                -- Insert the team association in the Reports table only if not already present
480                IF NOT EXISTS (SELECT 1 FROM Reports WHERE team_name = @team_name AND Employee_SSN = @SSN)
481                BEGIN
482                    INSERT INTO Reports (team_name, Employee_SSN, report_date, description)
483                    VALUES (@team_name, @SSN, @report_date, @description);
484                END
485            END TRY
486            BEGIN CATCH
487                PRINT 'Error inserting into Reports table for team: ' + @team_name;
488            END CATCH;
489        END
490        ELSE
491        BEGIN
492            PRINT 'Team not found: ' + @team_name;
493        END
494    END;
495 END;
496
497 select *from PAN;
498 select *from Teams;
499 select *from Reports;
500 GO

```

## Messages

---

```
3:58:00 AM    Started executing query at Line 419
               Commands completed successfully.

3:58:00 AM    Started executing query at Line 421
               Commands completed successfully.

               Total execution time: 00:00:00.128
```

### 6. Enter an expense charged by an employee (1/day).

```
502    -- Query 6
503    DROP PROCEDURE IF EXISTS EnterEmployeeExpense;
504    GO
505    CREATE PROCEDURE EnterEmployeeExpense
506        @Employee_SSN VARCHAR(20),
507        @description VARCHAR(255),
508        @expense_date DATE,
509        @amount DECIMAL(10, 2)
510    AS
511    BEGIN
512        -- Check if the employee exists
513        IF NOT EXISTS (SELECT 1 FROM Employee WHERE SSN = @Employee_SSN)
514        BEGIN
515            RAISERROR ('Employee not found with SSN: %s', 16, 1, @Employee_SSN);
516            RETURN;
517        END
518        -- Insert the expense charged by the employee
519        BEGIN TRY
520            INSERT INTO Expenses (Employee_SSN, description, expense_date, amount)
521            VALUES (@Employee_SSN, @description, @expense_date, @amount);
522            PRINT 'Expense successfully added for Employee SSN: ' + @Employee_SSN;
523        END TRY
524        BEGIN CATCH
525            DECLARE @ErrorMessage NVARCHAR(4000);
526            SET @ErrorMessage = ERROR_MESSAGE();
527            RAISERROR ('Error inserting expense for Employee SSN %s: %s', 16, 1, @Employee_SSN, @ErrorMessage);
528        END CATCH;
529    END;
530    GO
531    Select *from Employee;
532    select *from Expenses;
533    GO
```

---

Results

Messages

```
3:58:56 AM    Started executing query at Line 502
               Commands completed successfully.

3:58:56 AM    Started executing query at Line 505
               Commands completed successfully.

3:58:56 AM    Started executing query at Line 531
               (0 rows affected)
               (0 rows affected)
               Total execution time: 00:00:00.194
```

## 7. Enter a new donor and associate him or her with several donations (1/day).

```
534  --Query 7
535  DROP PROCEDURE IF EXISTS AddNewDonorWithDonations;
536  GO
537  CREATE PROCEDURE AddNewDonorWithDonations
538      @SSN VARCHAR(20),
539      @name VARCHAR(50),
540      @gender CHAR(1),
541      @profession VARCHAR(100),
542      @mailing_address VARCHAR(200),
543      @email_address VARCHAR(100),
544      @phone_no VARCHAR(15),
545      @mailing_list BIT,
546      @anonymous_donor BIT,
547      @donation_details NVARCHAR(MAX) -- Comma-separated list of donation details
548  AS
549  BEGIN
550      -- Insert into PAN if donor doesn't exist
551      IF NOT EXISTS (SELECT 1 FROM PAN WHERE SSN = @SSN)
552      BEGIN
553          INSERT INTO PAN (SSN, name, gender, profession, mailing_address, email_address, phone_no, mailing_list)
554          VALUES (@SSN, @name, @gender, @profession, @mailing_address, @email_address, @phone_no, @mailing_list);
555      END
556
557      -- Insert into Donor if not already present
558      IF NOT EXISTS (SELECT 1 FROM Donor WHERE SSN = @SSN)
559      BEGIN
560          INSERT INTO Donor (SSN, anonymous_donor)
561          VALUES (@SSN, @anonymous_donor);
562      END
563      -- Parse and process donations
```

```

563    -- Parse and process donations
564    DECLARE @donation_date DATE;
565    DECLARE @donation_amount DECIMAL(10, 2);
566    DECLARE @donation_type VARCHAR(50);
567    DECLARE @campaign_name VARCHAR(100);
568    DECLARE @cheque_number INT;
569    DECLARE @card_no VARCHAR(20);
570    DECLARE @card_type VARCHAR(20);
571    DECLARE @expiry_date DATE;
572
573    DECLARE @donation NVARCHAR(255);
574    DECLARE @pos INT;
575
576    WHILE CHARINDEX(';', @donation_details) > 0
577    BEGIN
578        SET @pos = CHARINDEX(';', @donation_details);
579        SET @donation = LTRIM(RTRIM(SUBSTRING(@donation_details, 1, @pos - 1)));
580        SET @donation_details = SUBSTRING(@donation_details, @pos + 1, LEN(@donation_details));
581
582        -- Parse each field from the donation details string based on commas
583        SET @donation_date = CAST(SUBSTRING(@donation, 1, CHARINDEX(',', @donation) - 1) AS DATE);
584        SET @donation = SUBSTRING(@donation, CHARINDEX(',', @donation) + 1, LEN(@donation));
585
586        SET @donation_amount = CAST(SUBSTRING(@donation, 1, CHARINDEX(',', @donation) - 1) AS DECIMAL(10, 2));
587        SET @donation = SUBSTRING(@donation, CHARINDEX(',', @donation) + 1, LEN(@donation));
588
589        SET @donation_type = LTRIM(RTRIM(SUBSTRING(@donation, 1, CHARINDEX(',', @donation) - 1)));
590        SET @donation = SUBSTRING(@donation, CHARINDEX(',', @donation) + 1, LEN(@donation));
591
592
593    IF LOWER(@donation_type) = 'cheque'
594    BEGIN
595        SET @cheque_number = CAST(@donation AS INT);
596
597        IF NOT EXISTS (SELECT 1 FROM Cheque WHERE cheque_number = @cheque_number)
598        BEGIN
599            INSERT INTO Cheque (Donor_SSN, donation_date, donation_amount, donation_type, campaign_name, cheque_number)
600            VALUES (@SSN, @donation_date, @donation_amount, @donation_type, @campaign_name, @cheque_number);
601        END
602    END
603
604    ELSE IF LOWER(@donation_type) = 'credit'
605    BEGIN
606        SET @card_no = LTRIM(RTRIM(SUBSTRING(@donation, 1, CHARINDEX(',', @donation) - 1)));
607        SET @donation = SUBSTRING(@donation, CHARINDEX(',', @donation) + 1, LEN(@donation));
608
609        SET @card_type = LTRIM(RTRIM(SUBSTRING(@donation, 1, CHARINDEX(',', @donation) - 1)));
610        SET @donation = SUBSTRING(@donation, CHARINDEX(',', @donation) + 1, LEN(@donation));
611
612        SET @expiry_date = CAST(@donation AS DATE);
613
614        IF NOT EXISTS (SELECT 1 FROM Credit_card WHERE card_no = @card_no)
615        BEGIN
616            INSERT INTO Credit_card (Donor_SSN, donation_date, donation_amount, donation_type, campaign_name, card_no, card_type, expiry_date)
617            VALUES (@SSN, @donation_date, @donation_amount, @donation_type, @campaign_name, @card_no, @card_type, @expiry_date);
618        END
619    END
620
621    ELSE
622    BEGIN
623        PRINT 'Invalid donation type: ' + @donation_type + '. Only "cheque" and "credit" are allowed.';
624        RETURN;
625    END
626
627
628    END
629    END;
630    GO
631    select *from PAN;
632    select *from Donor;
633    select *from Cheque;
634    select *from Credit_card;
635    GO

```

**Results      Messages**

---

```
4:01:18 AM    Started executing query at Line 535
               Commands completed successfully.
4:01:18 AM    Started executing query at Line 537
               Commands completed successfully.
4:01:18 AM    Started executing query at Line 629
               (0 rows affected)
               (0 rows affected)
               (0 rows affected)
               (0 rows affected)
               Total execution time: 00:00:00.315
```

**8. Retrieve the name and phone number of the doctor of a particular client (1/week).**

```
635  --Query 8
636  DROP PROCEDURE IF EXISTS GetClientDoctorInfo;
637  GO
638  CREATE PROCEDURE GetClientDoctorInfo
639  |   @Client_SSN VARCHAR(10)
640  AS
641  BEGIN
642  |   -- Retrieve the doctor's name and phone number for the specified client
643  |   SELECT doctor_name, doctor_phone_no
644  |   FROM Client
645  |   WHERE SSN = @Client_SSN;
646  END;
647  select * from Client;
648  GO
```

**Messages**

---

```
4:05:10 AM    Started executing query at Line 636
               Commands completed successfully.
4:05:10 AM    Started executing query at Line 638
               Commands completed successfully.
               Total execution time: 00:00:00.144
```

**9. 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 (1/month).**

```

-- 
650  --Query 9
651  select *from Employee;
652  select *from Expenses;
653  DROP PROCEDURE IF EXISTS GetTotalExpensesByEmployee;
654  GO
655
656  CREATE PROCEDURE GetTotalExpensesByEmployee
657      @start_date DATE,
658      @end_date DATE
659  AS
660  BEGIN
661      -- Retrieve the total amount of expenses charged by each employee within the specified period
662  SELECT
663      Employee_SSN,
664      SUM(amount) AS total_expenses
665  FROM
666      Expenses
667  WHERE
668      expense_date BETWEEN @start_date AND @end_date
669  GROUP BY
670      Employee_SSN
671  ORDER BY
672      total_expenses DESC;
673 END;
674 GO

```

Results	Messages
	4:07:10 AM      Started executing query at Line 651 (0 rows affected) (0 rows affected)
	4:07:10 AM      Started executing query at Line 655 Commands completed successfully. Total execution time: 00:00:00.135

**10. Retrieve the list of volunteers that are members of teams that support a particular client (4/year).**

```

676    --Query 10
677    DROP PROCEDURE IF EXISTS GetVolunteersForClient;
678    GO
679
680    CREATE PROCEDURE GetVolunteersForClient
681        @Client_SSN VARCHAR(10)
682    AS
683    BEGIN
684        SELECT Volunteer.SSN, Volunteer.training_location
685        FROM Volunteer
686        INNER JOIN Contain ON Volunteer.SSN = Contain.Volunteer_SSN
687        INNER JOIN Works ON Contain.team_name = Works.team_name
688        WHERE Works.Client_SSN = @Client_SSN;
689    END;
690    GO

```

Results	Messages
	4:07:46 AM <u>Started executing query at Line 677</u>
	Commands completed successfully.
	4:07:46 AM <u>Started executing query at Line 679</u>
	Commands completed successfully.
	4:07:46 AM <u>Started executing query at Line 691</u>
	(0 rows affected)
	Total execution time: 00:00:00.192

**11. Retrieve the names of all teams that were founded after a particular date (1/month).**

```

-- --
696 --Query 11
697 DROP PROCEDURE IF EXISTS GetTeamsFoundedAfterDate;
698 GO
699 CREATE PROCEDURE GetTeamsFoundedAfterDate
700     @date DATE
701 AS
702 BEGIN
703     -- Retrieve the names of all teams that were founded after the specified date
704     SELECT team_name
705     FROM Teams
706     WHERE team_date > @date;
707 END;
708 select *FROM Teams;
709 GO
710

```

---

## Messages

```

4:10:02 AM      Started executing query at Line 696
                  Commands completed successfully.
4:10:02 AM      Started executing query at Line 699
                  Commands completed successfully.
                  Total execution time: 00:00:00.146

```

## 12. 12. Retrieve the names, social security numbers, contact information, and emergency contact information of all people in the database (1/week).

```

711 --Query 12
712 DROP PROCEDURE IF EXISTS GetAllPeopleWithEmergencyContacts;
713 GO
714 CREATE PROCEDURE GetAllPeopleWithEmergencyContacts
715 AS
716 BEGIN
717     -- Retrieve names, SSNs, and contact information along with emergency contact information
718     SELECT
719         PAN.SSN,
720         PAN.name,
721         PAN.phone_no AS main_phone,
722         PAN.email_address,
723         Emergency_contact.phone_no AS emergency_phone,
724         Emergency_contact.name AS emergency_contact_name,
725         Emergency_contact.relation AS emergency_relation
726     FROM PAN
727     LEFT JOIN Emergency_contact ON PAN.SSN = Emergency_contact.SSN;
728 END;
729 select *from PAN;
730 GO

```

---

## Messages

```

4:11:09 AM      Started executing query at Line 712
                  Commands completed successfully.
4:11:09 AM      Started executing query at Line 714
                  Commands completed successfully.
                  Total execution time: 00:00:00.138

```

## 13. Retrieve the name and total amount donated by donors 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 (1/week).**

```
731  --Query 13
732  DROP PROCEDURE IF EXISTS GetEmployeeDonorTotalDonations;
733  GO
734  CREATE PROCEDURE GetEmployeeDonorTotalDonations
735  AS
736  BEGIN
737      -- Retrieve the name, total donation amount, and anonymous status for donors who are also employees
738      SELECT
739          PAN.name,
740          Donor.SSN,
741          SUM(COALESCE(Cheque.donation_amount, 0) + COALESCE(Credit_card.donation_amount, 0)) AS total_donations,
742          Donor.anonymous_donor
743      FROM Donor
744      INNER JOIN Employee ON Donor.SSN = Employee.SSN
745      INNER JOIN PAN ON Donor.SSN = PAN.SSN
746      LEFT JOIN Cheque ON Donor.SSN = Cheque.Donor_SSN
747      LEFT JOIN Credit_card ON Donor.SSN = Credit_card.Donor_SSN
748      GROUP BY PAN.name, Donor.SSN, Donor.anonymous_donor
749      ORDER BY total_donations DESC;
750  END;
751  select * from Cheque;
752  GO
753
```

#### Messages

```
4:12:10 AM    Started executing query at Line 732
               Commands completed successfully.
4:12:10 AM    Started executing query at Line 734
               Commands completed successfully.
               Total execution time: 00:00:00.153
```

**14. Increase the salary by 10% of all employees to whom more than one team must report. (1/year)**

```

754  --Query 14
755  DROP PROCEDURE IF EXISTS IncreaseSalaryForMultiTeamEmployees;
756  GO
757  CREATE PROCEDURE IncreaseSalaryForMultiTeamEmployees
758  AS
759  BEGIN
760      -- Update salaries for employees who are associated with more than one team
761      UPDATE Employee
762      SET emp_sal = emp_sal * 1.10
763      WHERE SSN IN (
764          SELECT Employee_SSN
765          FROM Reports
766          GROUP BY Employee_SSN
767          HAVING COUNT(DISTINCT team_name) > 1
768      );
769  END;
770  select *from Reports;
771  select *from Employee;
772  GO

```

## Messages

```

4:13:05 AM  Started executing query at Line 755
Commands completed successfully.
4:13:05 AM  Started executing query at Line 757
Commands completed successfully.
Total execution time: 00:00:00.239

```

## 15. Delete all clients who do not have health insurance and whose value of importance for transportation is less than 5 (4/year).

```

... --
773  --Query 15
774  DROP PROCEDURE IF EXISTS DeleteClientsWithoutHealthInsuranceAndLowTransportImportance;
775  GO
776  CREATE PROCEDURE DeleteClientsWithoutHealthInsuranceAndLowTransportImportance
777  AS
778  BEGIN
779      -- Temporary table to store the SSNs of clients to delete
780      DECLARE @clientsToDelete TABLE (SSN VARCHAR(20));
781      -- Populate the temporary table with clients who meet the deletion criteria
782      INSERT INTO @clientsToDelete (SSN)
783      SELECT c.SSN
784      FROM Client AS c
785      LEFT JOIN Insurance_policy AS i ON c.SSN = i.Client_SSN AND i.type = 'health'
786      LEFT JOIN Needs AS n ON c.SSN = n.Client_SSN AND n.needs_type = 'transportation'
787      WHERE i.Client_SSN IS NULL -- No health insurance
788      AND (n.importance_value IS NULL OR n.importance_value < 5); -- Low or missing transportation importance
789      -- Delete from dependent tables to avoid foreign key constraints
790      DELETE FROM Works WHERE Client_SSN IN (SELECT SSN FROM @clientsToDelete);
791      DELETE FROM Needs WHERE Client_SSN IN (SELECT SSN FROM @clientsToDelete);
792      DELETE FROM Insurance_policy WHERE Client_SSN IN (SELECT SSN FROM @clientsToDelete);
793      -- Finally, delete the clients from the Client table
794      DELETE FROM Client WHERE SSN IN (SELECT SSN FROM @clientsToDelete);
795      PRINT 'Clients deleted successfully based on the specified criteria.';
796  END;
797  select * from Client;
798  select *from Insurance_policy;
799  select *from Needs;
800  select *from Works;
801  GO

```

## 5.2. The Java source program

### Queries Method calling and display Menu

```
24      // Main loop to display menu and execute queries
25      while (!option.equals("18")) {
26          displayMenu();
27          option = sc.nextLine();
28
29          switch (option) {
30              case "1": addNewTeam(sc); break;
31              case "2": addNewClientWithTeams(sc); break;
32              case "3": addNewVolunteerWithTeams(sc); break;
33              case "4": enterVolunteerHours(sc); break;
34              case "5": addNewEmployeeWithTeams(sc); break;
35              case "6": enterEmployeeExpense(sc); break;
36              case "7": addNewDonorWithDonations(sc); break;
37              case "8": getClientDoctorInfo(sc); break;
38              case "9": getTotalExpensesByEmployee(sc); break;
39              case "10": getVolunteersForClient(sc); break;
40              case "11": getTeamsFoundedAfterDate(sc); break;
41              case "12": getAllPeopleWithEmergencyContacts(); break;
42              case "13": getEmployeeDonorTotalDonations(); break;
43              case "14": increaseSalaryForMultiTeamEmployees(); break;
44              case "15": deleteClientsWithoutHealthInsuranceAndLowTransportImportance(); break;
45              case "16": importData(); break;
46              case "17": exportData(); break;
47              case "18": System.out.println("Exiting program."); break;
48              default: System.out.println("Invalid option. Please try again."); break;
49          }
50      }
51      sc.close();
52  }
53  private static void displayMenu() {
54      System.out.println("\nSelect an option: ");
55      System.out.println("1) Add new team");
56      System.out.println("2) Add new client and associate with teams");
57      System.out.println("3) Add new volunteer and associate with teams");
58      System.out.println("4) Enter volunteer hours");
59      System.out.println("5) Add new employee and associate with teams");
60      System.out.println("6) Enter employee expense");
61      System.out.println("7) Add new donor with donations");
62      System.out.println("8) Get client doctor information");
63      System.out.println("9) Get total expenses by employee");
64      System.out.println("10) Get volunteers for client");
65      System.out.println("11) Get teams founded after a date");
66      System.out.println("12) Get all people with emergency contacts");
67      System.out.println("13) Get employee donors total donations");
68      System.out.println("14) Increase salary for multi-team employees");
69      System.out.println("15) Delete clients without health insurance and low transport importance");
70      System.out.println("16) Import data from file");
71      System.out.println("17) Export data to file");
72      System.out.println("18) Quit");
73  }
```

1.

```

75 // 1. Add a new team
76 private static void addNewTeam(Scanner sc) throws SQLException {
77     System.out.print("Enter team name: ");
78     String teamName = sc.next();
79     System.out.print("Enter team type: ");
80     String teamType = sc.next();
81     System.out.print("Enter team date (yyyy-mm-dd): ");
82     String teamDate = sc.next();
83
84     try (Connection conn = DriverManager.getConnection(URL)) {
85         String sql = "{call AddNewTeam(?, ?, ?)}";
86         try (CallableStatement stmt = conn.prepareCall(sql)) {
87             stmt.setString(1, teamName);
88             stmt.setString(2, teamType);
89             stmt.setDate(3, java.sql.Date.valueOf(teamDate));
90             stmt.execute();
91             System.out.println("New team added.");
92         }
93     }
94 }
```

## 2.

```

96 // 2. Add a new client and associate with teams
97 private static void addNewClientWithTeams(Scanner sc) throws SQLException {
98     System.out.print("Enter client SSN: ");
99     String ssn = sc.next();
100    sc.nextLine();
101    System.out.print("Enter client name: ");
102    String name = sc.nextLine();
103    System.out.print("Enter gender (M/F): ");
104    String gender = sc.next();
105    sc.nextLine();
106    System.out.print("Enter profession: ");
107    String profession = sc.nextLine();
108    System.out.print("Enter mailing address: ");
109    String address = sc.nextLine();
110    System.out.print("Enter email: ");
111    String email = sc.nextLine();
112    System.out.print("Enter phone number: ");
113    String phone = sc.nextLine();
114    System.out.print("Is the client subscribed to the mailing list? (true/false): ");
115    boolean mailingList = sc.nextBoolean();
116    sc.nextLine();
117    System.out.print("Enter doctor name: ");
118    String doctorName = sc.nextLine();
119    System.out.print("Enter doctor phone number: ");
120    String doctorPhone = sc.nextLine();
121    System.out.print("Enter date assigned (yyyy-mm-dd): ");
122    String dateAssigned = sc.next();
123    sc.nextLine();
124    System.out.print("Enter team names (comma-separated): ");
125    String teamNames = sc.nextLine();
126    System.out.print("Is the client active? (true/false): ");
127    boolean activeStatus = sc.nextBoolean();
```

```

125
126     try (Connection conn = DriverManager.getConnection(URL)) {
127         String sql = "{call AddNewClientWithTeams(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)}";
128         try (CallableStatement stmt = conn.prepareCall(sql)) {
129             stmt.setString(1, ssn);
130             stmt.setString(2, name);
131             stmt.setString(3, gender);
132             stmt.setString(4, profession);
133             stmt.setString(5, address);
134             stmt.setString(6, email);
135             stmt.setString(7, phone);
136             stmt.setBoolean(8, mailingList);
137             stmt.setString(9, doctorName);
138             stmt.setString(10, doctorPhone);
139             stmt.setDate(11, java.sql.Date.valueOf(dateAssigned));
140             stmt.setString(12, teamNames);
141             stmt.setBoolean(13, activeStatus); // Set the active/inactive status
142
143             stmt.execute();
144             System.out.println("New client and team associations added.");
145         }
146     }
147 }
148

```

### 3.

```

152 // 3. Add a new volunteer and associate with teams
153 private static void addNewVolunteerWithTeams(Scanner sc) throws SQLException {
154     System.out.print("Enter volunteer SSN: ");
155     String ssn = sc.next();
156     sc.nextLine();
157     System.out.print("Enter volunteer name: ");
158     String name = sc.next();
159     sc.nextLine();
160     System.out.print("Enter gender (M/F): ");
161     String gender = sc.next();
162     sc.nextLine();
163     System.out.print("Enter profession: ");
164     String profession = sc.next();
165     sc.nextLine();
166     System.out.print("Enter mailing address: ");
167     String address = sc.next();
168     sc.nextLine();
169     System.out.print("Enter email: ");
170     String email = sc.next();
171     sc.nextLine();
172     System.out.print("Enter phone number: ");
173     String phone = sc.next();
174     sc.nextLine();
175     System.out.print("Is the volunteer subscribed to the mailing list? (true/false): ");
176     boolean mailingList = sc.nextBoolean();
177     System.out.print("Enter date joined (yyyy-mm-dd): ");
178     sc.nextLine();
179     String dateJoined = sc.next();
180     System.out.print("Enter training date (yyyy-mm-dd): ");
181     sc.nextLine();
182     String trainingDate = sc.next();
183

```

```

182     String trainingDate = sc.next();
183     System.out.print("Enter training location: ");
184     sc.nextLine();
185     String trainingLocation = sc.next();
186     sc.nextLine();
187     System.out.print("Enter team names (comma-separated): ");
188     String teamNames = sc.next();
189     System.out.print("Is the volunteer active? (true/false): ");
190     boolean activeStatus = sc.nextBoolean();
191
192     try (Connection conn = DriverManager.getConnection(URL)) {
193
194         String sql = "{call AddNewVolunteerWithTeams(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)}";
195         try (CallableStatement stmt = conn.prepareCall(sql)) {
196             stmt.setString(1, ssn);
197             stmt.setString(2, name);
198             stmt.setString(3, gender);
199             stmt.setString(4, profession);
200             stmt.setString(5, address);
201             stmt.setString(6, email);
202             stmt.setString(7, phone);
203             stmt.setBoolean(8, mailingList);
204             stmt.setDate(9, java.sql.Date.valueOf(dateJoined));
205             stmt.setDate(10, java.sql.Date.valueOf(trainingDate));
206             stmt.setString(11, trainingLocation);
207             stmt.setString(12, teamNames);
208             stmt.setBoolean(13, activeStatus); // Set the active/inactive status
209             stmt.execute();
210             System.out.println("New volunteer and team associations added.");
211         }
212     }
213 }
```

#### 4.

```

202 // 4. Enter volunteer hours
203 private static void enterVolunteerHours(Scanner sc) throws SQLException {
204     System.out.print("Enter team name: ");
205     String teamName = sc.next();
206     System.out.print("Enter volunteer SSN: ");
207     String volunteerSSN = sc.next();
208     System.out.print("Enter hours worked: ");
209     int hours = sc.nextInt();
210     System.out.print("Enter month (Jan, Feb, etc.): ");
211     String month = sc.next();
212
213     try (Connection conn = DriverManager.getConnection(URL)) {
214         String sql = "{call EnterVolunteerHours(?, ?, ?, ?)}";
215         try (CallableStatement stmt = conn.prepareCall(sql)) {
216             stmt.setString(1, teamName);
217             stmt.setString(2, volunteerSSN);
218             stmt.setInt(3, hours);
219             stmt.setString(4, month);
220             stmt.execute();
221             System.out.println("Volunteer hours recorded.");
222         }
223     }
224 }
225 }
```

## 5.

```
239 // 5. Add new employee and associate with teams
240 private static void addNewEmployeeWithTeams(Scanner sc) throws SQLException {
241     System.out.print("Enter Employee SSN: ");
242     String ssn = sc.next();
243     sc.nextLine();
244     System.out.print("Enter employee name: ");
245     String name = sc.next();
246     sc.nextLine();
247     System.out.print("Enter gender (M/F): ");
248     String gender = sc.next();
249     sc.nextLine();
250     System.out.print("Enter profession: ");
251     String profession = sc.next();
252     sc.nextLine();
253     System.out.print("Enter mailing address: ");
254     String address = sc.next();
255     sc.nextLine();
256     System.out.print("Enter email: ");
257     String email = sc.next();
258     sc.nextLine();
259     System.out.print("Enter phone number: ");
260     String phone = sc.next();
261     sc.nextLine();
262     System.out.print("Is the Employee subscribed to the mailing list? (true/false): ");
263     boolean mailingList = sc.nextBoolean();
264     sc.nextLine();
265     System.out.print("Enter employee salary: ");
266     double salary = sc.nextDouble();
267     sc.nextLine();
268     System.out.print("Enter marital status: ");
269     String maritalStatus = sc.next();
270     ...
271 
272     sc.nextLine();
273     System.out.print("Enter hire date (yyyy-mm-dd): ");
274     String hireDate = sc.next();
275     sc.nextLine();
276     System.out.print("Enter team names (comma-separated): ");
277     String teamNames = sc.next();
278     sc.nextLine();
279     System.out.print("Enter report date (yyyy-mm-dd): ");
280     String report_date = sc.next();
281     sc.nextLine();
282     System.out.print("Enter description");
283     String description = sc.next();
284 
285     try (Connection conn = DriverManager.getConnection(URL)) {
286         String sql = "{call AddNewEmployeeWithTeams(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)}";
287         try (CallableStatement stmt = conn.prepareCall(sql)) {
288             stmt.setString(1, ssn);
289             stmt.setString(2, name);
290             stmt.setString(3, gender);
291             stmt.setString(4, profession);
292             stmt.setString(5, address);
293             stmt.setString(6, email);
294             stmt.setString(7, phone);
295             stmt.setBoolean(8, mailingList);
296             stmt.setDouble(9, salary);
297             stmt.setString(10, maritalStatus);
298             stmt.setDate(11, java.sql.Date.valueOf(hireDate));
299             stmt.setString(12, teamNames);
300             stmt.setString(13, report_date);
301             stmt.setString(14, description);
302             stmt.execute();
303             System.out.println("Employee added successfully.");
304         }
305     }
```

```

300         stmt.execute();
301         System.out.println("New employee and team associations added.");
302     }
303 }
304 ---
```

## 6.

```

281 // 6. Enter employee expense
282 private static void enterEmployeeExpense(Scanner sc) throws SQLException {
283     System.out.print("Enter Employee SSN: ");
284     String ssn = sc.next();
285     System.out.print("Enter expense description: ");
286     String description = sc.next();
287     System.out.print("Enter expense date (yyyy-mm-dd): ");
288     String expenseDate = sc.next();
289     System.out.print("Enter expense amount: ");
290     double amount = sc.nextDouble();
291
292     try (Connection conn = DriverManager.getConnection(URL)) {
293         String sql = "{call EnterEmployeeExpense(?, ?, ?, ?)}";
294         try (CallableStatement stmt = conn.prepareCall(sql)) {
295             stmt.setString(1, ssn);
296             stmt.setString(2, description);
297             stmt.setDate(3, java.sql.Date.valueOf(expenseDate));
298             stmt.setDouble(4, amount);
299             stmt.execute();
300             System.out.println("Employee expense recorded.");
301         }
302     }
303 }
```

## 7.

```

305 // 7. Add new donor with donations
306 private static void addNewDonorWithDonations(Scanner sc) throws SQLException {
307     System.out.print("Enter Donor SSN: ");
308     String ssn = sc.next();
309     sc.nextLine(); // Clear buffer
310     System.out.print("Enter donor name: ");
311     String name = sc.nextLine();
312     System.out.print("Enter gender (M/F): ");
313     String gender = sc.next();
314     sc.nextLine(); // Clear buffer after single-word input
315     System.out.print("Enter profession: ");
316     String profession = sc.nextLine();
317     System.out.print("Enter mailing address: ");
318     String address = sc.nextLine();
319     System.out.print("Enter email: ");
320     String email = sc.nextLine();
321     System.out.print("Enter phone number: ");
322     String phone = sc.nextLine();
323     System.out.print("Is the donor subscribed to the mailing list? (true/false): ");
324     boolean mailingList = sc.nextBoolean();
325     System.out.print("Is the donor anonymous (1 for yes, 0 for no): ");
326     int anonymous = sc.nextInt();
327     sc.nextLine(); // Clear buffer after integer input
328
329     StringBuilder donationDetails = new StringBuilder();
330     boolean addMore = true;
331
332     while (addMore) {
333         System.out.print("Enter donation type (cheque/credit): ");
334         String donationType = sc.nextLine().trim();
```

```

336     double donationAmount = 0;
337     boolean validAmount = false;
338
339     while (!validAmount) {
340         System.out.print("Enter donation amount: ");
341         String amountInput = sc.nextLine().trim(); // Read the entire line
342         try {
343             donationAmount = Double.parseDouble(amountInput); // Try parsing the amount as double
344             validAmount = true; // Set flag if parsing succeeds
345         } catch (NumberFormatException e) {
346             System.out.println("Invalid amount. Please enter a valid numeric value.");
347         }
348     }
349
350     System.out.print("Enter donation date (yyyy-mm-dd): ");
351     String donationDate = sc.nextLine();
352     System.out.print("Enter campaign name: ");
353     String campaignName = sc.nextLine();
354
355     if ("cheque".equalsIgnoreCase(donationType)) {
356         System.out.print("Enter cheque number: ");
357         int chequeNumber = sc.nextInt();
358         sc.nextLine(); // Clear buffer
359         donationDetails.append(donationDate).append(",").append(donationAmount).append(",")
360             .append("cheque,").append(campaignName).append(",").append(chequeNumber).append(";");
361     } else if ("credit".equalsIgnoreCase(donationType)) {
362         System.out.print("Enter card number: ");
363         String cardNumber = sc.next();
364         System.out.print("Enter card type: ");
365         String cardType = sc.next();
366         System.out.print("Enter expiry date (yyyy-mm-dd): ");
367         String expiryDate = sc.next();
368
369         System.out.print("Enter expiry date (yyyy-mm-dd): ");
370         String expiryDate = sc.next();
371         sc.nextLine(); // Clear buffer after credit card details
372         donationDetails.append(donationDate).append(",").append(donationAmount).append(",")
373             .append("credit,").append(campaignName).append(",").append(cardNumber)
374             .append(",").append(cardType).append(",").append(expiryDate).append(";");
375     } else {
376         System.out.println("Invalid donation type entered: " + donationType + ". Only 'Cheque'"
377             + " and 'Credit Card' are allowed.");
378         continue;
379     }
380
381     System.out.print("Do you want to add another donation? (yes/no): ");
382     addMore = sc.nextLine().trim().equalsIgnoreCase("yes");
383 }
384
385 try (Connection conn = DriverManager.getConnection(URL)) {
386     String sql = "{call AddNewDonorWithDonations(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)}";
387     try (CallableStatement stmt = conn.prepareCall(sql)) {
388         stmt.setString(1, ssn);
389         stmt.setString(2, name);
390         stmt.setString(3, gender);
391         stmt.setString(4, profession);
392         stmt.setString(5, address);
393         stmt.setString(6, email);
394         stmt.setString(7, phone);
395         stmt.setBoolean(8, mailingList);
396         stmt.setInt(9, anonymous);
397         stmt.setString(10, donationDetails.toString());
398         stmt.execute();
399         System.out.println("New donor with donations recorded.");
400     }
401 }

```

8.

```

401 // 8. Get client doctor information
402 private static void getClientDoctorInfo(Scanner sc) throws SQLException {
403     System.out.print("Enter Client SSN: ");
404     String clientSSN = sc.next();
405
406     try (Connection conn = DriverManager.getConnection(URL)) {
407         String sql = "{call GetClientDoctorInfo(?)}";
408         try (CallableStatement stmt = conn.prepareCall(sql)) {
409             stmt.setString(1, clientSSN);
410             try (ResultSet rs = stmt.executeQuery()) {
411                 if (rs.next()) {
412                     System.out.println("Doctor Name: " + rs.getString("doctor_name"));
413                     System.out.println("Doctor Phone: " + rs.getString("doctor_phone_no"));
414                 } else {
415                     System.out.println("No doctor information found for the provided client SSN.");
416                 }
417             }
418         }
419     }
420 }
421

```

## 9.

```

-- //
422 // 9. Get total expenses by employee within a specific period
423 private static void getTotalExpensesByEmployee(Scanner sc) throws SQLException {
424     System.out.print("Enter start date (yyyy-mm-dd): ");
425     String startDate = sc.next();
426     System.out.print("Enter end date (yyyy-mm-dd): ");
427     String endDate = sc.next();
428
429     try (Connection conn = DriverManager.getConnection(URL)) {
430         String sql = "{call GetTotalExpensesByEmployee(?, ?)}";
431         try (CallableStatement stmt = conn.prepareCall(sql)) {
432             stmt.setDate(1, java.sql.Date.valueOf(startDate));
433             stmt.setDate(2, java.sql.Date.valueOf(endDate));
434             try (ResultSet rs = stmt.executeQuery()) {
435                 System.out.println("Employee SSN | Total Expenses");
436                 while (rs.next()) {
437                     System.out.printf("%s | %.2f%n", rs.getString("Employee_SSN"), rs.getDouble("total_expenses"));
438                 }
439             }
440         }
441     }
442 }
443

```

## 10.

```

444
445 // 10. Get volunteers for a specific client
446 private static void getVolunteersForClient(Scanner sc) throws SQLException {
447     System.out.print("Enter Client SSN: ");
448     String clientSSN = sc.next();
449
450     try (Connection conn = DriverManager.getConnection(URL)) {
451         String sql = "{call GetVolunteersForClient(?)}";
452         try (CallableStatement stmt = conn.prepareCall(sql)) {
453             stmt.setString(1, clientSSN);
454             try (ResultSet rs = stmt.executeQuery()) {
455                 System.out.println("Volunteer SSN | Training Location");
456                 while (rs.next()) {
457                     System.out.printf("%s | %s%n", rs.getString("SSN"), rs.getString("training_location"));
458                 }
459             }
460         }
461     }
462 }
463

```

## 11.

```

464 // 11. Get teams founded after a specific date
465 private static void getTeamsFoundedAfterDate(Scanner sc) throws SQLException {
466     System.out.print("Enter date (yyyy-mm-dd): ");
467     String date = sc.next();
468
469     try (Connection conn = DriverManager.getConnection(URL)) {
470         String sql = "{call GetTeamsFoundedAfterDate(?)}";
471         try (CallableStatement stmt = conn.prepareCall(sql)) {
472             stmt.setDate(1, java.sql.Date.valueOf(date));
473             try (ResultSet rs = stmt.executeQuery()) {
474                 System.out.println("Teams founded after " + date + ":");
475                 while (rs.next()) {
476                     System.out.println("Team Name: " + rs.getString("team_name"));
477                 }
478             }
479         }
480     }
481 }

```

## 12.

```

483 // 12. Get all people with their emergency contacts
484 private static void getAllPeopleWithEmergencyContacts() throws SQLException {
485     try (Connection conn = DriverManager.getConnection(URL)) {
486         String sql = "{call GetAllPeopleWithEmergencyContacts}";
487         try (CallableStatement stmt = conn.prepareCall(sql); ResultSet rs = stmt.executeQuery()) {
488             System.out.println("SSN | Name | Main Phone | Email | Emergency Phone | Emergency Contact Name | Relation");
489             while (rs.next()) {
490                 System.out.printf("%s | %s | %s | %s | %s | %s | %s%n",
491                     rs.getString("SSN"),
492                     rs.getString("name"),
493                     rs.getString("main_phone"),
494                     rs.getString("email_address"),
495                     rs.getString("emergency_phone"),
496                     rs.getString("emergency_contact_name"),
497                     rs.getString("emergency_relation"));
498             }
499         }
500     }
501 }
502

```

## 13.

```

503 // 13. Get total donations from employee donors
504 private static void getEmployeeDonorTotalDonations() throws SQLException {
505     try (Connection conn = DriverManager.getConnection(URL)) {
506         String sql = "{call GetEmployeeDonorTotalDonations}";
507         try (CallableStatement stmt = conn.prepareCall(sql);
508              ResultSet rs = stmt.executeQuery()) {
509             System.out.println("Name | SSN | Total Donations | Anonymous Donor");
510             while (rs.next()) {
511                 System.out.printf("%s | %s | %.2f | %s%n",
512                     rs.getString("name"), rs.getString("SSN"), rs.getDouble("total_donations"),
513                     rs.getBoolean("anonymous_donor") ? "Yes" : "No");
514             }
515         }
516     }
517 }
518
519

```

## 14.

```

519
520     // 14. Increase salary for employees associated with multiple teams
521     private static void increaseSalaryForMultiTeamEmployees() throws SQLException {
522         try (Connection conn = DriverManager.getConnection(URL)) {
523             String sql = "{call IncreaseSalaryForMultiTeamEmployees}";
524             try (CallableStatement stmt = conn.prepareCall(sql)) {
525                 stmt.execute();
526                 System.out.println("Salary increased by 10% for eligible employees.");
527             }
528         }
529     }
530 }

```

## 15.

```

531     // 15. Delete clients without health insurance and low transport importance
532     private static void deleteClientsWithoutHealthInsuranceAndLowTransportImportance() throws SQLException {
533         try (Connection conn = DriverManager.getConnection(URL)) {
534             String sql = "{call DeleteClientsWithoutHealthInsuranceAndLowTransportImportance}";
535             try (CallableStatement stmt = conn.prepareCall(sql)) {
536                 stmt.execute();
537                 System.out.println("Clients without health insurance and low transportation importance deleted.");
538             }
539         }
540     }
541 ...

```

## 16. Import: enter new teams from a data file until the file is empty (the user must be asked to enter the input file name).

```

541 //16 import
542     private static void importData() throws SQLException {
543         Scanner sc = new Scanner(System.in);
544         System.out.print("Enter the input file name: ");
545         String fileName = sc.nextLine();
546
547         try (BufferedReader br = new BufferedReader(new FileReader(fileName))) {
548             String line;
549             Connection conn = DriverManager.getConnection(URL);
550
551             while ((line = br.readLine()) != null) {
552                 String[] teamData = line.split(",");
553                 if (teamData.length == 3) {
554                     String teamName = teamData[0].trim();
555                     String teamType = teamData[1].trim();
556                     Date teamDate = Date.valueOf(teamData[2].trim());
557
558                     // Insert each team into the Teams table
559                     String sql = "{call AddNewTeam(?, ?, ?)}";
560                     try (CallableStatement stmt = conn.prepareCall(sql)) {
561                         stmt.setString(1, teamName);
562                         stmt.setString(2, teamType);
563                         stmt.setDate(3, teamDate);
564                         stmt.execute();
565                         System.out.println("Team added: " + teamName);
566                     } catch (SQLException e) {
567                         System.err.println("Failed to add team: " + teamName + ". Error: " + e.getMessage());
568                     }
569                 } else {
570                     System.err.println("Invalid line format: " + line);
571                 }
572             }
573         }
574     }
575 }

```

```

570             System.err.println("Invalid line format: " + line);
571         }
572     }
573     conn.close();
574     System.out.println("Import completed.");
575
576 } catch (IOException e) {
577     System.err.println("Error reading file: " + e.getMessage());
578 }
579
580

```

**17. Export:** Retrieve names and mailing addresses of all people on the mailing list and output them to a data file instead of screen (the user must be asked to enter the output file name).

```

~~~
582 //17 Sample implementation for exporting data to CSV (for `exportData`)
583 private static void exportData() throws SQLException {
584     Scanner sc = new Scanner(System.in);
585     System.out.print("Enter the output file name: ");
586     String fileName = sc.nextLine();
587
588     try (Connection conn = DriverManager.getConnection(URL)) {
589         String sql = "SELECT name, mailing_address FROM PAN WHERE mailing_list = 1";
590         try (PreparedStatement stmt = conn.prepareStatement(sql));
591             ResultSet rs = stmt.executeQuery() {
592
593                 try (BufferedWriter writer = new BufferedWriter(new FileWriter(fileName))) {
594                     writer.write("Name, Mailing Address\n"); // Header for CSV
595                     while (rs.next()) {
596                         String name = rs.getString("name");
597                         String mailingAddress = rs.getString("mailing_address");
598                         writer.write(name + ", " + mailingAddress + "\n");
599                     }
600                     System.out.println("Mailing list exported to " + fileName);
601                 } catch (IOException e) {
602                     System.err.println("Error writing to file: " + e.getMessage());
603                 }
604             }
605         }
606     }
607 }

```

## Task 6: Java program Execution

### 6.1. Screenshots showing the testing of query 1

```
Individual_project [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (Nov 14, 2024, 4:32:59 AM) [pid: 12616]
Welcome to the PAN Database Application!

Select an option:
1) Add new team
2) Add new client and associate with teams
3) Add new volunteer and associate with teams
4) Enter volunteer hours
5) Add new employee and associate with teams
6) Enter employee expense
7) Add new donor with donations
8) Get client doctor information
9) Get total expenses by employee
10) Get volunteers for client
11) Get teams founded after a date
12) Get all people with emergency contacts
13) Get employee donors total donations
14) Increase salary for multi-team employees
15) Delete clients without health insurance and low transport importance
16) Import data from file
17) Export data to file
18) Quit
1
Enter team name: A-team
Enter team type: sports
Enter team date (yyyy-mm-dd): 2023-01-13
New team added.
```

```
Individual_project [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (Nov 14, 2024, 4:32:59 AM) [pid: 12616]

Select an option:
1) Add new team
2) Add new client and associate with teams
3) Add new volunteer and associate with teams
4) Enter volunteer hours
5) Add new employee and associate with teams
6) Enter employee expense
7) Add new donor with donations
8) Get client doctor information
9) Get total expenses by employee
10) Get volunteers for client
11) Get teams founded after a date
12) Get all people with emergency contacts
13) Get employee donors total donations
14) Increase salary for multi-team employees
15) Delete clients without health insurance and low transport importance
16) Import data from file
17) Export data to file
18) Quit
1
Enter team name: B-Team
Enter team type: swimming
Enter team date (yyyy-mm-dd): 2024-12-23
New team added.
```

Individual\_project [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (Nov 14, 2024, 4:32:59 AM) [pid: 12616]

```
Select an option:  
1) Add new team  
2) Add new client and associate with teams  
3) Add new volunteer and associate with teams  
4) Enter volunteer hours  
5) Add new employee and associate with teams  
6) Enter employee expense  
7) Add new donor with donations  
8) Get client doctor information  
9) Get total expenses by employee  
10) Get volunteers for client  
11) Get teams founded after a date  
12) Get all people with emergency contacts  
13) Get employee donors total donations  
14) Increase salary for multi-team employees  
15) Delete clients without health insurance and low transport importance  
16) Import data from file  
17) Export data to file  
18) Quit
```

1

```
Enter team name: C-Team  
Enter team type: singing  
Enter team date (yyyy-mm-dd): 2022-03-12  
New team added.
```

Individual\_project [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (Nov 14, 2024, 4:32:59 AM) [pid: 12616]

```
Select an option:  
1) Add new team  
2) Add new client and associate with teams  
3) Add new volunteer and associate with teams  
4) Enter volunteer hours  
5) Add new employee and associate with teams  
6) Enter employee expense  
7) Add new donor with donations  
8) Get client doctor information  
9) Get total expenses by employee  
10) Get volunteers for client  
11) Get teams founded after a date  
12) Get all people with emergency contacts  
13) Get employee donors total donations  
14) Increase salary for multi-team employees  
15) Delete clients without health insurance and low transport importance  
16) Import data from file  
17) Export data to file  
18) Quit
```

1

```
Enter team name: D-Team  
Enter team type: dancing  
Enter team date (yyyy-mm-dd): 2021-03-13  
New team added.
```

Individual\_project [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (Nov 14, 2024, 4:32:59 AM) [pid: 12616]

Select an option:  
1) Add new team  
2) Add new client and associate with teams  
3) Add new volunteer and associate with teams  
4) Enter volunteer hours  
5) Add new employee and associate with teams  
6) Enter employee expense  
7) Add new donor with donations  
8) Get client doctor information  
9) Get total expenses by employee  
10) Get volunteers for client  
11) Get teams founded after a date  
12) Get all people with emergency contacts  
13) Get employee donors total donations  
14) Increase salary for multi-team employees  
15) Delete clients without health insurance and low transport importance  
16) Import data from file  
17) Export data to file  
18) Quit

1

Enter team name: E-Team  
Enter team type: Debat  
Enter team date (yyyy-mm-dd): 2019-06-24  
New team added.

#### Output:

##### Affected Teams Table:

```
221      LIND,  
222      SELECT * FROM Teams; -- for testing  
223      GO
```

##### Results    Messages

	team_name	team_type	team_date
1	A-team	sports	2023-01-13
2	B-Team	swimming	2024-12-23
3	C-Team	singing	2022-03-12
4	D-Team	dancing	2021-03-13
5	E-Team	Debat	2019-06-24

## 6.2. Screenshots showing the testing of query 2

```
Individual_project [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (Nov 14, 2024, 4:56:58 AM) [pid: 8808]
Welcome to the PAN Database Application!

Select an option:
1) Add new team
2) Add new client and associate with teams
3) Add new volunteer and associate with teams
4) Enter volunteer hours
5) Add new employee and associate with teams
6) Enter employee expense
7) Add new donor with donations
8) Get client doctor information
9) Get total expenses by employee
10) Get volunteers for client
11) Get teams founded after a date
12) Get all people with emergency contacts
13) Get employee donors total donations
14) Increase salary for multi-team employees
15) Delete clients without health insurance and low transport importance
16) Import data from file
17) Export data to file
18) Quit
2
Enter client SSN: 1234567890
Enter client name: Feroz
Enter gender (M/F): M
Enter profession: Software
Enter mailing address: 123 Eastern Avenue
Enter email: feroz@gmail.com
Enter phone number: 4056783245
Is the client subscribed to the mailing list? (true/false): true
Enter doctor name: Tyson
```

```
Enter doctor phone number: 4537892390
Enter date assigned (yyyy-mm-dd): 2022-06-9
Enter team names (comma-separated): A-Team,C-Team
Is the client active? (true/false): false
New client and team associations added.
```

Select an option:

- 1) Add new team
- 2) Add new client and associate with teams
- 3) Add new volunteer and associate with teams
- 4) Enter volunteer hours
- 5) Add new employee and associate with teams
- 6) Enter employee expense
- 7) Add new donor with donations
- 8) Get client doctor information
- 9) Get total expenses by employee
- 10) Get volunteers for client
- 11) Get teams founded after a date
- 12) Get all people with emergency contacts
- 13) Get employee donors total donations
- 14) Increase salary for multi-team employees
- 15) Delete clients without health insurance and low transport importance
- 16) Import data from file
- 17) Export data to file
- 18) Quit

2

```
Enter client SSN: 2345678901
```

```
Enter client name: Jackson
```

---

```
Individual_project [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (Nov 14, 2024, 4:56:58AM) [pid: 8808]
Enter client name: Jackson
Enter gender (M/F): M
Enter profession: SInger
Enter mailing address: 234 Preston Woods
Enter email: jackson123@gmail.com
Enter phone number: 4563217894
Is the client subscribed to the mailing list? (true/false): true
Enter doctor name: Nikhil
Enter doctor phone number: 6543278904
Enter date assigned (yyyy-mm-dd): 2023-03-4
Enter team names (comma-separated): D-Team,
Is the client active? (true/false): true
New client and team associations added.

Select an option:
1) Add new team
2) Add new client and associate with teams
3) Add new volunteer and associate with teams
4) Enter volunteer hours
5) Add new employee and associate with teams
6) Enter employee expense
7) Add new donor with donations
8) Get client doctor information
9) Get total expenses by employee
10) Get volunteers for client
11) Get teams founded after a date
12) Get all people with emergency contacts
13) Get employee donors total donations
14) Increase salary for multi-team employees
15) Delete clients without health insurance and low transport importance
16) Import data from file
```

```
Individual_project [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (Nov 14, 2024, 4:56:58 AM) [pid: 8808]
17) Export data to file
18) Quit
2
Enter client SSN: 3456789012
Enter client name: Nigel
Enter gender (M/F): M
Enter profession: Teacher
Enter mailing address: 214 parkwood street
Enter email: nigel1324@gmail.com
Enter phone number: 5673289458
Is the client subscribed to the mailing list? (true/false): true
Enter doctor name: Magaratte
Enter doctor phone number: 6547893456
Enter date assigned (yyyy-mm-dd): 2019-05-22
Enter team names (comma-separated): B-Team,
Is the client active? (true/false): false
New client and team associations added.

Select an option:
1) Add new team
2) Add new client and associate with teams
3) Add new volunteer and associate with teams
4) Enter volunteer hours
5) Add new employee and associate with teams
6) Enter employee expense
7) Add new donor with donations
8) Get client doctor information
9) Get total expenses by employee
10) Get volunteers for client
11) Get teams founded after a date
```

```
11) Get teams founded after a date
12) Get all people with emergency contacts
13) Get employee donors total donations
14) Increase salary for multi-team employees
15) Delete clients without health insurance and low transport importance
16) Import data from file
17) Export data to file
18) Quit
2
Enter client SSN: 4567890123
Enter client name: Darin
Enter gender (M/F): M
Enter profession: Dancer
Enter mailing address: 220 Deonne Circle
Enter email: darin452@gmail.com
Enter phone number: 5467237922
Is the client subscribed to the mailing list? (true/false): false
Enter doctor name: Kiran
Enter doctor phone number: 4748383905
Enter date assigned (yyyy-mm-dd): 2010-08-11
Enter team names (comma-separated): E-Team,D-Team,B-Team
Is the client active? (true/false): true
New client and team associations added.

Select an option:
1) Add new team
2) Add new client and associate with teams
3) Add new volunteer and associate with teams
4) Enter volunteer hours
5) Add new employee and associate with teams
6) Enter employee expense
```

---

```

6) Enter employee expense
7) Add new donor with donations
8) Get client doctor information
9) Get total expenses by employee
10) Get volunteers for client
11) Get teams founded after a date
12) Get all people with emergency contacts
13) Get employee donors total donations
14) Increase salary for multi-team employees
15) Delete clients without health insurance and low transport importance
16) Import data from file
17) Export data to file
18) Quit
2
Enter client SSN: 5678901234
Enter client name: Indira
Enter gender (M/F): F
Enter profession: Singer
Enter mailing address: 532 eastern street
Enter email: indira54@gmail.com
Enter phone number: 4748304093
Is the client subscribed to the mailing list? (true/false): true
Enter doctor name: Nag
Enter doctor phone number: 4638439740
Enter date assigned (yyyy-mm-dd): 2018-09-9
Enter team names (comma-separated): A-Team,
Is the client active? (true/false): true
New client and team associations added.

```

Select an option:

1) Add new team

---

#### Output:

#### Affected Tables (Client,Works,PAN)

291    `SELECT * from Client;`

---

**Results**    **Messages**

	SSN	doctor_name	doctor_phone_no	date_assigned
1	1234567890	Tyson	4537892390	2022-06-09
2	2345678901	Nikhil	6543278904	2023-03-04
3	3456789012	Magaratte	6547893456	2019-05-22
4	4567890123	Kiran	4748383905	2010-08-11
5	5678901234	Nag	4638439740	2018-09-09

292     Select \* from Works;

Results     Messages

	team_name	client_SSN	active_inactive
1	A-Team	1234567890	0
2	A-Team	5678901234	1
3	B-Team	3456789012	0
4	B-Team	4567890123	1
5	C-Team	1234567890	0
6	D-Team	2345678901	1
7	D-Team	4567890123	1
8	E-Team	4567890123	1

293     Select \*from PAN;

Results     Messages

	SSN	name	gender	profession	mailing_address	email_address	phone_no	mailing_list
1	1234567890	Feroz	M	Software	123 Eastern Avenue	feroz@gmail.com	4056783245	1
2	2345678901	Jackson	M	Singer	234 Preston Woods	jackson123@gmail.com	4563217894	1
3	3456789012	Nigel	M	Teacher	214 parkwood street	nigel324@gmail.com	5673289458	1
4	4567890123	Darin	M	Dancer	220 Deonne Circle	darin452@gmail.com	5467237922	0
5	5678901234	Indira	F	Singer	532 eastern street	indira54@gmail.com	4748304093	1

### 6.3. Screenshots showing the testing of query 3

```
Individual_project [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (Nov 14, 2024, 9:17:43 AM) [pid: 12972]
Welcome to the PAN Database Application!

Select an option:
1) Add new team
2) Add new client and associate with teams
3) Add new volunteer and associate with teams
4) Enter volunteer hours
5) Add new employee and associate with teams
6) Enter employee expense
7) Add new donor with donations
8) Get client doctor information
9) Get total expenses by employee
10) Get volunteers for client
11) Get teams founded after a date
12) Get all people with emergency contacts
13) Get employee donors total donations
14) Increase salary for multi-team employees
15) Delete clients without health insurance and low transport importance
16) Import data from file
17) Export data to file
18) Quit
3
Enter volunteer SSN: 5678901347
Enter volunteer name: Adam Dixon
Enter gender (M/F): M
Enter profession: Student
Enter mailing address: 129 Apt Suite Riverwalk road
Enter email: adamdi@gmail.com
```

```
Individual_project [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (Nov 14, 2024, 9:17:43 AM) [pid: 12972]
```

```
Enter email: adamdi@gmail.com
Enter phone number: 4567829302
Is the volunteer subscribed to the mailing list? (true/false): true
Enter date joined (yyyy-mm-dd): 2017-09-22
Enter training date (yyyy-mm-dd): 2017-09-25
Enter training location: Univeristy of Oklahoma
Enter team names (comma-separated): A-Team,B-Team
Is the volunteer active? (true/false): true
New volunteer and team associations added.
```

```
Select an option:
1) Add new team
2) Add new client and associate with teams
3) Add new volunteer and associate with teams
4) Enter volunteer hours
5) Add new employee and associate with teams
6) Enter employee expense
7) Add new donor with donations
8) Get client doctor information
9) Get total expenses by employee
10) Get volunteers for client
11) Get teams founded after a date
12) Get all people with emergency contacts
13) Get employee donors total donations
14) Increase salary for multi-team employees
15) Delete clients without health insurance and low transport importance
16) Import data from file
17) Export data to file
18) Quit
```

```
3
```

```
Enter volunteer SSN: 6011002344
Enter volunteer name: Lyndi
Enter gender (M/F): F
Enter profession: Student
Enter mailing address: 145 Startbucks Coffee
Enter email: lyndi12@gmail.com
Enter phone number: 45672840234
```

```
Enter phone number: 45672840234
Is the volunteer subscribed to the mailing list? (true/false): true
Enter date joined (yyyy-mm-dd): 2017-04-23
Enter training date (yyyy-mm-dd): 2018-09-24
Enter training location: OU residence
Enter team names (comma-separated): C-Team,
Is the volunteer active? (true/false): true
New volunteer and team associations added.

Select an option:
1) Add new team
2) Add new client and associate with teams
3) Add new volunteer and associate with teams
4) Enter volunteer hours
5) Add new employee and associate with teams
6) Enter employee expense
7) Add new donor with donations
8) Get client doctor information
9) Get total expenses by employee
10) Get volunteers for client
11) Get teams founded after a date
12) Get all people with emergency contacts
13) Get employee donors total donations
14) Increase salary for multi-team employees
15) Delete clients without health insurance and low transport importance
16) Import data from file
17) Export data to file
18) Quit
```

```
10) Get volunteers for client
11) Get teams founded after a date
12) Get all people with emergency contacts
13) Get employee donors total donations
14) Increase salary for multi-team employees
15) Delete clients without health insurance and low transport importance
16) Import data from file
17) Export data to file
18) Quit
3
Enter volunteer SSN: 6011002409
Enter volunteer name: Sara
Enter gender (M/F): F
Enter profession: chef
Enter mailing address: 456 Preston wrail
Enter email: sara@gmail.com
Enter phone number: 5637848029
Is the volunteer subscribed to the mailing list? (true/false): false
Enter date joined (yyyy-mm-dd): 2016-04-11
Enter training date (yyyy-mm-dd): 2017-08-5
Enter training location: campus corner
Enter team names (comma-separated): D-Team,
Is the volunteer active? (true/false): false
New volunteer and team associations added.
```

Select an option:

1) Add new team

---

```
Select an option:
1) Add new team
2) Add new client and associate with teams
3) Add new volunteer and associate with teams
4) Enter volunteer hours
5) Add new employee and associate with teams
6) Enter employee expense
7) Add new donor with donations
8) Get client doctor information
9) Get total expenses by employee
10) Get volunteers for client
11) Get teams founded after a date
12) Get all people with emergency contacts
13) Get employee donors total donations
14) Increase salary for multi-team employees
15) Delete clients without health insurance and low transport importance
16) Import data from file
17) Export data to file
18) Quit
3
Enter volunteer SSN: 6011002456
Enter volunteer name: Cornner
Enter gender (M/F): M
Enter profession: Dancer
Enter mailing address: 356 Western blvd
Enter email: corner@gmail.com
Enter phone number: 5638932757
Is the volunteer subscribed to the mailing list? (true/false): false
Enter date joined (yyyy-mm-dd): 2011-07-3
Enter training date (yyyy-mm-dd): 2014-05-12
```

---

```
Enter training location: redstron
Enter team names (comma-separated): D-Team,E-Team,
Is the volunteer active? (true/false): false
New volunteer and team associations added.

Select an option:
1) Add new team
2) Add new client and associate with teams
3) Add new volunteer and associate with teams
4) Enter volunteer hours
5) Add new employee and associate with teams
6) Enter employee expense
7) Add new donor with donations
8) Get client doctor information
9) Get total expenses by employee
10) Get volunteers for client
11) Get teams founded after a date
12) Get all people with emergency contacts
13) Get employee donors total donations
14) Increase salary for multi-team employees
15) Delete clients without health insurance and low transport importance
16) Import data from file
17) Export data to file
18) Quit
3
Enter volunteer SSN: 6011003102
Enter volunteer name: Adam chadler
Enter gender (M/F): M
Enter profession: Singer
Enter mailing address: 3727 classen blvd

Enter email: adamc@gmail.com
Enter phone number: 4638749203
Is the volunteer subscribed to the mailing list? (true/false): true
Enter date joined (yyyy-mm-dd): 2022-04-22
Enter training date (yyyy-mm-dd): 2022-05-23
Enter training location: dreston
Enter team names (comma-separated): C-Team
Is the volunteer active? (true/false): true
New volunteer and team associations added.

Select an option:
1) Add new team
2) Add new client and associate with teams
3) Add new volunteer and associate with teams
4) Enter volunteer hours
5) Add new employee and associate with teams
6) Enter employee expense
7) Add new donor with donations
8) Get client doctor information
9) Get total expenses by employee
10) Get volunteers for client
11) Get teams founded after a date
12) Get all people with emergency contacts
13) Get employee donors total donations
14) Increase salary for multi-team employees
15) Delete clients without health insurance and low transport importance
16) Import data from file
17) Export data to file
18) Quit
```

Ouput:

Affected Tables(Volunteer,Contain, PAN)

364    `SELECT * from Volunteer;`

Results    Messages

	SSN	date_joined	training_date	training_location
1	5678901347	2017-09-22	2017-09-25	Univeristy
2	6011002344	2017-04-23	2018-09-24	OU
3	6011002409	2016-04-11	2017-08-05	campus
4	6011002456	2011-07-03	2014-05-12	redstron
5	6011003102	2022-04-22	2022-05-23	dreston

365    `Select * from Contain;`

Results    Messages

	team_name	Volunteer_SSN	active_inactive
1	A-Team	5678901347	1
2	B-Team	5678901347	1
3	C-Team	6011002344	1
4	C-Team	6011003102	1
5	D-Team	6011002409	0
6	D-Team	6011002456	0
7	E-Team	6011002456	0

366    `Select *from PAN;`

Results    Messages

	SSN	name	gender	profession	mailing_address	email_address	phone_no	mailing_list
1	1234567890	Feroz	M	Software	123 Eastern Avenue	feroz@gmail.com	4056783245	1
2	2345678901	Jackson	M	SInger	234 Preston Woods	jackson123@gmail.com	4563217894	1
3	3456789012	Nigel	M	Teacher	214 parkwood street	nigel324@gmail.com	5673289458	1
4	4567890123	Darin	M	Dancer	220 Deonne Circle	darin452@gmail.com	5467237922	0
5	5678901234	Indira	F	Singer	532 eastern street	indira54@gmail.com	4748304093	1
6	5678901347	Adam	M	Student	129	adamdi@gmail.com	4567829302	1
7	6011002344	Lyndi	F	Student	145	lyndi12@gmail.com	45672840234	1
8	6011002409	Sara	F	chef	456	sara@gmail.com	5637848029	0
9	6011002456	Cornner	M	Dancer	356	corner@gmail.com	5638932757	0
10	6011003102	Adam	M	Singer	3727	adamc@gmail.com	4638749203	1

#### 6.4. Screenshots showing the testing of query 4

18) Quit

4

```
Enter team name: A-Team
Enter volunteer SSN: 5678901347
Enter hours worked: 20
Enter month (Jan, Feb, etc.): Jan
Volunteer hours recorded.
```

Select an option:

- 1) Add new team
- 2) Add new client and associate with teams
- 3) Add new volunteer and associate with teams
- 4) Enter volunteer hours
- 5) Add new employee and associate with teams
- 6) Enter employee expense
- 7) Add new donor with donations
- 8) Get client doctor information
- 9) Get total expenses by employee
- 10) Get volunteers for client
- 11) Get teams founded after a date
- 12) Get all people with emergency contacts
- 13) Get employee donors total donations
- 14) Increase salary for multi-team employees
- 15) Delete clients without health insurance and low transport importance
- 16) Import data from file
- 17) Export data to file
- 18) Quit

4

4

Enter team name: B-Team  
Enter volunteer SSN: 5678901347  
Enter hours worked: 22  
Enter month (Jan, Feb, etc.): Feb  
Volunteer hours recorded.

Select an option:

- 1) Add new team
- 2) Add new client and associate with teams
- 3) Add new volunteer and associate with teams
- 4) Enter volunteer hours
- 5) Add new employee and associate with teams
- 6) Enter employee expense
- 7) Add new donor with donations
- 8) Get client doctor information
- 9) Get total expenses by employee
- 10) Get volunteers for client
- 11) Get teams founded after a date
- 12) Get all people with emergency contacts
- 13) Get employee donors total donations
- 14) Increase salary for multi-team employees
- 15) Delete clients without health insurance and low transport importance
- 16) Import data from file
- 17) Export data to file
- 18) Quit

```
10) Quit
4
Enter team name: C-Team
Enter volunteer SSN: 6011002344
Enter hours worked: 21
Enter month (Jan, Feb, etc.): Mar
Volunteer hours recorded.

Select an option:
1) Add new team
2) Add new client and associate with teams
3) Add new volunteer and associate with teams
4) Enter volunteer hours
5) Add new employee and associate with teams
6) Enter employee expense
7) Add new donor with donations
8) Get client doctor information
9) Get total expenses by employee
10) Get volunteers for client
11) Get teams founded after a date
12) Get all people with emergency contacts
13) Get employee donors total donations
14) Increase salary for multi-team employees
15) Delete clients without health insurance and low transport importance
16) Import data from file
17) Export data to file
18) Quit
```

```
4
Enter team name: D-Team
Enter volunteer SSN: 6011002409
Enter hours worked: 23
Enter month (Jan, Feb, etc.): Jul
Volunteer hours recorded.

Select an option:
1) Add new team
2) Add new client and associate with teams
3) Add new volunteer and associate with teams
4) Enter volunteer hours
5) Add new employee and associate with teams
5) Enter employee expense
7) Add new donor with donations
8) Get client doctor information
9) Get total expenses by employee
10) Get volunteers for client
11) Get teams founded after a date
12) Get all people with emergency contacts
13) Get employee donors total donations
14) Increase salary for multi-team employees
15) Delete clients without health insurance and low transport importance
16) Import data from file
17) Export data to file
18) Quit
```

```

4
Enter team name: D-Team
Enter volunteer SSN: 6011002409
Enter hours worked: 25
Enter month (Jan, Feb, etc.): Mar
Volunteer hours recorded.

Select an option:
1) Add new team
2) Add new client and associate with teams
3) Add new volunteer and associate with teams
4) Enter volunteer hours
5) Add new employee and associate with teams
6) Enter employee expense
7) Add new donor with donations
8) Get client doctor information
9) Get total expenses by employee
10) Get volunteers for client
11) Get teams founded after a date
12) Get all people with emergency contacts
13) Get employee donors total donations
14) Increase salary for multi-team employees
15) Delete clients without health insurance and low transport importance
16) Import data from file
17) Export data to file
18) Quit

```

**Output:**

**Affected Tables( Hour/Month)**

```

415      select * from Hour_month;
-- 20

```

**Results      Messages**

	team_name	Volunteer_SSN	hours	months
1	A-Team	5678901347	20	Jan
2	B-Team	5678901347	22	Feb
3	C-Team	6011002344	21	Mar
4	D-Team	6011002409	23	Jul
5	D-Team	6011002409	25	Mar

## 6.5. Screenshots showing the testing of query 5

Individual\_project [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (Nov 14, 2024, 9:54:16 AM) [pid: 21552]  
Welcome to the PAN Database Application!

Select an option:  
1) Add new team  
2) Add new client and associate with teams  
3) Add new volunteer and associate with teams  
4) Enter volunteer hours  
5) Add new employee and associate with teams  
6) Enter employee expense  
7) Add new donor with donations  
8) Get client doctor information  
9) Get total expenses by employee  
10) Get volunteers for client  
11) Get teams founded after a date  
12) Get all people with emergency contacts  
13) Get employee donors total donations  
14) Increase salary for multi-team employees  
15) Delete clients without health insurance and low transport importance  
16) Import data from file  
17) Export data to file  
18) Quit

5

Enter Employee SSN: 6011003203

Enter employee name: Dexon

Enter gender (M/F): F

Enter profession: Painter

Enter mailing address: 2356 redwen trails

Enter email: dexon@gmail.com

Enter phone number: 5623748038

Is the Employee subscribed to the mailing list? (true/false): true

Enter employee salary: 456783

Individual\_project [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (Nov 14, 2024, 9:54:16 AM) [pid: 21552]

```
Enter employee salary: 456783
Enter marital status: single
Enter hire date (yyyy-mm-dd): 2023-04-12
Enter team names (comma-separated): A-Team,B-Team,
Enter report date (yyyy-mm-dd): 2023-05-13
Enter description: description1
New employee and team associations added.
```

Select an option:

- 1) Add new team
- 2) Add new client and associate with teams
- 3) Add new volunteer and associate with teams
- 4) Enter volunteer hours
- 5) Add new employee and associate with teams
- 6) Enter employee expense
- 7) Add new donor with donations
- 8) Get client doctor information
- 9) Get total expenses by employee
- 10) Get volunteers for client
- 11) Get teams founded after a date
- 12) Get all people with emergency contacts
- 13) Get employee donors total donations
- 14) Increase salary for multi-team employees
- 15) Delete clients without health insurance and low transport importance
- 16) Import data from file
- 17) Export data to file
- 18) Quit

5

```
Enter Employee SSN: 6011014562
```

```
Enter employee name: Roshini
```

```
Enter gender (M/F): F
```

```
individual_project Java Application| C:\Program Files\Java\jdk-23\bin\javaw.exe (Nov 14, 2024, 9:54:16 AM) [pid: 2152]
Enter profession: Teacher
Enter mailing address: 931 Denonne Circle
Enter email: roshini@gmail.com
Enter phone number: 4705995025
Is the Employee subscribed to the mailing list? (true/false): true
Enter employee salary: 678904
Enter marital status: single
Enter hire date (yyyy-mm-dd): 2022-04-12
Enter team names (comma-separated): A-Team,D-Team,
Enter report date (yyyy-mm-dd): 2023-05-13
Enter descriptiondescription2
New employee and team associations added.

Select an option:
1) Add new team
2) Add new client and associate with teams
3) Add new volunteer and associate with teams
4) Enter volunteer hours
5) Add new employee and associate with teams
6) Enter employee expense
7) Add new donor with donations
8) Get client doctor information
9) Get total expenses by employee
10) Get volunteers for client
11) Get teams founded after a date
12) Get all people with emergency contacts
13) Get employee donors total donations
14) Increase salary for multi-team employees
15) Delete clients without health insurance and low transport importance
16) Import data from file
17) Export data to file
18) Quit
5
Enter Employee SSN: 6119390246
Enter employee name: Deepika
Enter gender (M/F): F
Enter profession: Scientist
Enter mailing address: 3452 drake drive
Enter email: deepika@gmail.com
Enter phone number: 5723980278
Is the Employee subscribed to the mailing list? (true/false): false
Enter employee salary: 6484498
Enter marital status: single
Enter hire date (yyyy-mm-dd): 2012-04-03
Enter team names (comma-separated): C-Team,D-Team,E-Team,
Enter report date (yyyy-mm-dd): 2015-04-02
Enter descriptiondescription 3
New employee and team associations added.
```

Select an option:

- 1) Add new team
- 2) Add new client and associate with teams
- 3) Add new volunteer and associate with teams
- 4) Enter volunteer hours
- 5) Add new employee and associate with teams
- 6) Enter employee expense
- 7) Add new donor with donations
- 8) Get client doctor information
- 9) Get total expenses by employee
- 10) Get volunteers for client
- 11) Get teams founded after a date
- 12) Get all people with emergency contacts
- 13) Get employee donors total donations
- 14) Increase salary for multi-team employees
- 15) Delete clients without health insurance and low transport importance
- 16) Import data from file
- 17) Export data to file
- 18) Quit

5

Enter Employee SSN: 6124363754  
Enter employee name: Reena  
Enter gender (M/F): f  
Enter profession: flightattendant  
Enter mailing address: 3432 treeware street  
Enter email: reena24@gmail.com  
Enter phone number: 3528639278  
Is the Employee subscribed to the mailing list? (true/false): true  
Enter employee salary: 363879  
Enter marital status: single

```
Enter hire date (yyyy-mm-dd): 2014-04-23
Enter team names (comma-separated): A-Team,C-Team
Enter report date (yyyy-mm-dd):2015-03-26
Enter descriptiondescription4
New employee and team associations added.

Select an option:
1) Add new team
2) Add new client and associate with teams
3) Add new volunteer and associate with teams
4) Enter volunteer hours
5) Add new employee and associate with teams
6) Enter employee expense
7) Add new donor with donations
8) Get client doctor information
9) Get total expenses by employee
10) Get volunteers for client
11) Get teams founded after a date
12) Get all people with emergency contacts
13) Get employee donors total donations
14) Increase salary for multi-team employees
15) Delete clients without health insurance and low transport importance
16) Import data from file
17) Export data to file
18) Quit
5

5
Enter Employee SSN: 6281392353
Enter employee name: Ava
Enter gender (M/F): F
Enter profession: rider
Enter mailing address: 5376 drake street
Enter email: avak@gmail.com
Enter phone number: 5768373833
Is the Employee subscribed to the mailing list? (true/false): false
Enter employee salary: 45367
Enter marital status: single
Enter hire date (yyyy-mm-dd): 2011-3-23
Enter team names (comma-separated): A-Team,
Enter report date (yyyy-mm-dd):2015-03-25
Enter descriptiondescription5
New employee and team associations added.
```

Output:

Affected Tables(Employee ,Reports and PAN)

```

500    select *from Employee;
501    select *from Reports;

```

**Results**    **Messages**

	SSN	emp_sal	marital_status	hire_date
1	6011003203	456783.00	single	2023-04-12
2	6011014562	678904.00	single	2022-04-12
3	6119390246	6484498.00	single	2012-04-03
4	6124363754	363879.00	single	2014-04-23
5	6281392353	45367.00	single	2011-03-23

	team_name	Employee_SSN	report_date	description
1	A-Team	6011003203	2023-05-13	description1
2	B-Team	6011003203	2023-05-13	description1
3	C-Team	6119390246	2015-04-02	description
4	D-Team	6011014562	2023-05-13	description2
5	E-Team	6119390246	2015-04-02	description

**Results**    **Messages**

	SSN	name	gender	profession	mailing_address	email_address	phone_no	mailing_list
1	1234567890	Feroz	M	Software	123 Eastern Avenue	feroz@gmail.com	4056783245	1
2	2345678901	Jackson	M	Singer	234 Preston Woods	jackson123@gmail.com	4563217894	1
3	3456789012	Nigel	M	Teacher	214 parkwood street	nigel324@gmail.com	5673289458	1
4	4567890123	Darin	M	Dancer	220 Deonne Circle	darin452@gmail.com	5467237922	0
5	5678901234	Indira	F	Singer	532 eastern street	indira54@gmail.com	4748304093	1
6	5678901347	Adam	M	Student	129	adamdi@gmail.com	4567829302	1
7	6011002344	Lyndi	F	Student	145	lyndi12@gmail.com	45672840234	1
8	6011002409	Sara	F	chef	456	sara@gmail.com	5637848029	0
9	6011002456	Cornner	M	Dancer	356	corner@gmail.com	5638932757	0
10	6011003102	Adam	M	Singer	3727	adamc@gmail.com	4638749203	1
11	6011003203	Dexon	F	Painter	2356	dexon@gmail.com	5623748038	1
12	6011014562	Roshini	F	Teacher	931	roshini@gmail.com	4705995025	1
13	6119390246	Deepika	F	Scientist	3452	deepika@gmail.com	5723980278	0
14	6124363754	Reena	f	flightattenda...	3432	reena24@gmail.com	3528639278	1
15	6281392353	Ava	F	rider	5376	avak@gmail.com	5768373833	0

## 6.6. Screenshots showing the testing of query 6

Select an option:

- 1) Add new team
- 2) Add new client and associate with teams
- 3) Add new volunteer and associate with teams
- 4) Enter volunteer hours
- 5) Add new employee and associate with teams
- 6) Enter employee expense
- 7) Add new donor with donations
- 8) Get client doctor information
- 9) Get total expenses by employee
- 10) Get volunteers for client
- 11) Get teams founded after a date
- 12) Get all people with emergency contacts
- 13) Get employee donors total donations
- 14) Increase salary for multi-team employees
- 15) Delete clients without health insurance and low transport importance
- 16) Import data from file
- 17) Export data to file
- 18) Quit

6

Enter Employee SSN: 6119390246

Enter expense description: birthday

Enter expense date (yyyy-mm-dd): 2023-05-23

Enter expense amount: 200000

Employee expense recorded.

Select an option:

- 1) Add new team
  - 2) Add new client and associate with teams
-

```
3) Add new volunteer and associate with teams
4) Enter volunteer hours
5) Add new employee and associate with teams
6) Enter employee expense
7) Add new donor with donations
8) Get client doctor information
9) Get total expenses by employee
10) Get volunteers for client
11) Get teams founded after a date
12) Get all people with emergency contacts
13) Get employee donors total donations
14) Increase salary for multi-team employees
15) Delete clients without health insurance and low transport importance
16) Import data from file
17) Export data to file
18) Quit
6
Enter Employee SSN: 6119390246
Enter expense description: cab
Enter expense date (yyyy-mm-dd): 2024-04-13
Enter expense amount: 2456
Employee expense recorded.
```

```
Select an option:
1) Add new team
2) Add new client and associate with teams
3) Add new volunteer and associate with teams
4) Enter volunteer hours
5) Add new employee and associate with teams
6) Enter employee expense
```

- 
- 6) Enter employee expense
  - 7) Add new donor with donations
  - 8) Get client doctor information
  - 9) Get total expenses by employee
  - 10) Get volunteers for client
  - 11) Get teams founded after a date
  - 12) Get all people with emergency contacts
  - 13) Get employee donors total donations
  - 14) Increase salary for multi-team employees
  - 15) Delete clients without health insurance and low transport importance
  - 16) Import data from file
  - 17) Export data to file
  - 18) Quit

**6**

Enter Employee SSN: **6124363754**

Enter expense description: **medical**

Enter expense date (yyyy-mm-dd): **2024-03-23**

Enter expense amount: **254673**

Employee expense recorded.

Select an option:

- 1) Add new team
- 2) Add new client and associate with teams
- 3) Add new volunteer and associate with teams
- 4) Enter volunteer hours
- 5) Add new employee and associate with teams
- 6) Enter employee expense
- 7) Add new donor with donations
- 8) Get client doctor information
- 9) Get total expenses by employee
- 10) Get volunteers for client

- 
- 10) Get volunteers for client
  - 11) Get teams founded after a date
  - 12) Get all people with emergency contacts
  - 13) Get employee donors total donations
  - 14) Increase salary for multi-team employees
  - 15) Delete clients without health insurance and low transport importance
  - 16) Import data from file
  - 17) Export data to file
  - 18) Quit

6

Enter Employee SSN: 6011014562  
Enter expense description: medical  
Enter expense date (yyyy-mm-dd): 2013-04-3  
Enter expense amount: 2345  
Employee expense recorded.

Select an option:

- 1) Add new team
- 2) Add new client and associate with teams
- 3) Add new volunteer and associate with teams
- 4) Enter volunteer hours
- 5) Add new employee and associate with teams
- 6) Enter employee expense
- 7) Add new donor with donations
- 8) Get client doctor information
- 9) Get total expenses by employee
- 10) Get volunteers for client
- 11) Get teams founded after a date
- 12) Get all people with emergency contacts
- 13) Get employee donors total donations
- 14) Increase salary for multi-team employees

```

individual_project.java Application C:\Program Files\Java\jdk-23\bin\javaw.exe (NOV 14, 2024, 12:35:43 PM) [pi]
15) Delete clients without health insurance and low transport importance
16) Import data from file
17) Export data to file
18) Quit
6
Enter Employee SSN: 6281392353
Enter expense description: benefits
Enter expense date (yyyy-mm-dd): 2018-09-23
Enter expense amount: 35672
Employee expense recorded.

Select an option:
1) Add new team
2) Add new client and associate with teams
3) Add new volunteer and associate with teams
4) Enter volunteer hours
5) Add new employee and associate with teams
6) Enter employee expense
7) Add new donor with donations
8) Get client doctor information
9) Get total expenses by employee
10) Get volunteers for client
11) Get teams founded after a date
12) Get all people with emergency contacts
13) Get employee donors total donations
14) Increase salary for multi-team employees
15) Delete clients without health insurance and low transport importance
16) Import data from file
17) Export data to file
18) Quit

```

#### Output:

##### Affected Tables(Expense)

```

533      SELECT * FROM Employee,
534      select *from Expenses;
535      GO

```

##### Results      Messages

	Employee_SSN	description	expense_date	amount
1	6011014562	medical	2013-04-03	2345.00
2	6119390246	birthday	2023-05-23	200000.00
3	6119390246	cab	2024-04-13	2456.00
4	6124363754	medical	2024-03-23	254673.00
5	6281392353	benefits	2018-09-23	35672.00

#### 6.7. Screenshots showing the testing of query 7

---

Select an option:

- 1) Add new team
- 2) Add new client and associate with teams
- 3) Add new volunteer and associate with teams
- 4) Enter volunteer hours
- 5) Add new employee and associate with teams
- 6) Enter employee expense
- 7) Add new donor with donations
- 8) Get client doctor information
- 9) Get total expenses by employee
- 10) Get volunteers for client
- 11) Get teams founded after a date
- 12) Get all people with emergency contacts
- 13) Get employee donors total donations
- 14) Increase salary for multi-team employees
- 15) Delete clients without health insurance and low transport importance
- 16) Import data from file
- 17) Export data to file
- 18) Quit

7

Enter Donor SSN: 6281392467

Enter donor name: Jai

Enter gender (M/F): M

Enter profession: Cricketer

Enter mailing address: 123 weston street

Enter email: jai2@gmail.com

Enter phone number: 5749803229

Is the donor subscribed to the mailing list? (true/false): true

Is the donor anonymous (1 for yes, 0 for no): 1

Enter donation type (cheque/credit): cheque

Enter donation amount: 557820

---

```
Enter donation type (cheque/cash): cheque
Enter donation amount: 567839
Enter donation date (yyyy-mm-dd): 2019-01-02
Enter campaign name: campaign1
Enter cheque number: 45672389
Do you want to add another donation? (yes/no): no
New donor with donations recorded.
```

Select an option:

- 1) Add new team
- 2) Add new client and associate with teams
- 3) Add new volunteer and associate with teams
- 4) Enter volunteer hours
- 5) Add new employee and associate with teams
- 6) Enter employee expense
- 7) Add new donor with donations
- 8) Get client doctor information
- 9) Get total expenses by employee
- 10) Get volunteers for client
- 11) Get teams founded after a date
- 12) Get all people with emergency contacts
- 13) Get employee donors total donations
- 14) Increase salary for multi-team employees
- 15) Delete clients without health insurance and low transport importance
- 16) Import data from file
- 17) Export data to file
- 18) Quit

7

---

---

7

```
Enter Donor SSN: 6281456789
Enter donor name: Srikanth
Enter gender (M/F): M
Enter profession: Chef
Enter mailing address: 123 classen blvd
Enter email: sri@gmail.com
Enter phone number: 4739238297
Is the donor subscribed to the mailing list? (true/false): true
Is the donor anonymous (1 for yes, 0 for no): 0
Enter donation type (cheque/credit): credit
Enter donation amount: 45678
Enter donation date (yyyy-mm-dd): 2019-04-23
Enter campaign name: campaign2
Enter card number: 4483799328367894
Enter card type: visa
Enter expiry date (yyyy-mm-dd): 2027-04-23
Do you want to add another donation? (yes/no): no
New donor with donations recorded.
```

Select an option:

- 1) Add new team
  - 2) Add new client and associate with teams
  - 3) Add new volunteer and associate with teams
  - 4) Enter volunteer hours
  - 5) Add new employee and associate with teams
  - 6) Enter employee expense
  - 7) Add new donor with donations
  - 8) Get client doctor information
  - 9) Get total expenses by employee
  - 10) Get volunteers for client
-

---

```
13) Get employee donors total donations
14) Increase salary for multi-team employees
15) Delete clients without health insurance and low transport importance
16) Import data from file
17) Export data to file
18) Quit
7
Enter Donor SSN: 6381557890
Enter donor name: Julie
Enter gender (M/F): F
Enter profession: Doctor
Enter mailing address: 463 western lake
Enter email: juile@gmail.com
Enter phone number: 4684739839
Is the donor subscribed to the mailing list? (true/false): false
Is the donor anonymous (1 for yes, 0 for no): 1
Enter donation type (cheque/credit): cheque
Enter donation amount: 583749
Enter donation date (yyyy-mm-dd): 2019-03-12
Enter campaign name: campaign3
Enter cheque number: 46493
Do you want to add another donation? (yes/no): yes
Enter donation type (cheque/credit): credit
Enter donation amount: 64837
Enter donation date (yyyy-mm-dd): 2020-12-23
Enter campaign name: campaign3
Enter card number: 6478373829382373
Enter card type: master
Enter expiry date (yyyy-mm-dd): 2030-03-24
Do you want to add another donation? (yes/no): no
New donor with donations recorded.
```

```
Select an option:  
1) Add new team  
2) Add new client and associate with teams  
3) Add new volunteer and associate with teams  
4) Enter volunteer hours  
5) Add new employee and associate with teams  
6) Enter employee expense  
7) Add new donor with donations  
8) Get client doctor information  
9) Get total expenses by employee  
10) Get volunteers for client  
11) Get teams founded after a date  
12) Get all people with emergency contacts  
13) Get employee donors total donations  
14) Increase salary for multi-team employees  
15) Delete clients without health insurance and low transport importance  
16) Import data from file  
17) Export data to file  
18) Quit  
7  
Enter Donor SSN: 6437882899  
Enter donor name: Sreeja  
Enter gender (M/F): F  
Enter profession: writer  
Enter mailing address: 939 deonne circle  
Enter email: sreeja@gmail.com  
Enter phone number: 5647308409  
Is the donor subscribed to the mailing list? (true/false): true  
Is the donor anonymous (1 for yes, 0 for no): 1  
Enter donation type (cheque/credit): cheque  
Enter donation amount: 75495
```

```
Enter donation amount: 75495
Enter donation date (yyyy-mm-dd): 2019-04-23
Enter campaign name: campaign2
Enter cheque number: 47347
Do you want to add another donation? (yes/no): no
New donor with donations recorded.
```

Select an option:

- 1) Add new team
- 2) Add new client and associate with teams
- 3) Add new volunteer and associate with teams
- 4) Enter volunteer hours
- 5) Add new employee and associate with teams
- 6) Enter employee expense
- 7) Add new donor with donations
- 8) Get client doctor information
- 9) Get total expenses by employee
- 10) Get volunteers for client
- 11) Get teams founded after a date
- 12) Get all people with emergency contacts
- 13) Get employee donors total donations
- 14) Increase salary for multi-team employees
- 15) Delete clients without health insurance and low transport importance
- 16) Import data from file
- 17) Export data to file
- 18) Quit

```
7
```

```
Enter Donor SSN: 6583938838
Enter donor name: Teja
Enter gender (M/F): M
Enter profession: Engineer
Enter mailing address: 2200 classen blvd
Enter email: teja@gmail.com
Enter phone number: 6483984038
Is the donor subscribed to the mailing list? (true/false): true
Is the donor anonymous (1 for yes, 0 for no): 0
Enter donation type (cheque/credit): credit
Enter donation amount: 36733
Enter donation date (yyyy-mm-dd): 2018-02-12
Enter campaign name: campaign2
Enter card number: 4634738032397858
Enter card type: rupay
Enter expiry date (yyyy-mm-dd): 2032-04-12
Do you want to add another donation? (yes/no): no
New donor with donations recorded.
```

## Output

### Affected Tables(Donor, Cheque, Credit\_card, PAN)

Results Messages

	SSN	anonymous_donor
1	6281392467	1
2	6281456789	0
3	6381557890	1
4	6437882899	1
5	6583938838	0

	Donor_SSN	donation_date	donation_amount	donation_type	campaign_name	cheque_number
1	6281392467	2019-01-02	567839.00	cheque	campaign1	45672389
2	6381557890	2019-03-12	583749.00	cheque	campaign3	46493
3	6437882899	2019-04-23	75495.00	cheque	campaign2	47347

	Donor_SSN	donation_date	donation_amount	donation_type	campaign_name	card_no	card_type	expiry_date
1	6281456789	2019-04-23	45678.00	credit	campaign2	4483799328367894	visa	2027-04-23
2	6381557890	2020-12-23	64837.00	credit	campaign3	6478373829382373	master	2030-03-24
3	6583938838	2018-02-12	36733.00	credit	campaign2	4634738032397858	rupay	2032-04-12

### PAN Table(added donors successfully)

16	6281392467	Jai	M	Cricketer	123 weston street	jai2@gmail.com	5749803229	1
17	6281456789	Srikan...	M	Chef	123 classen blvd	sri@gmail.com	4739238297	1
18	6381557890	Julie	F	Doctor	463 western lake	juile@gmail.com	4684739839	0
19	6437882899	Sreeja	F	writer	939 deonne circle	sreeja@gmail.com	5647308409	1
20	6583938838	Teja	M	Engineer	2200 classen blvd	teja@gmail.com	6483984038	1

### 6.8. Screenshots showing the testing of query 8

```

8
Enter Client SSN: 2345678901
Doctor Name: Nikhil
Doctor Phone: 6543278904

Select an option:
1) Add new team
2) Add new client and associate with teams
3) Add new volunteer and associate with teams
4) Enter volunteer hours
5) Add new employee and associate with teams
6) Enter employee expense
7) Add new donor with donations
8) Get client doctor information
9) Get total expenses by employee
10) Get volunteers for client
11) Get teams founded after a date
12) Get all people with emergency contacts
13) Get employee donors total donations
14) Increase salary for multi-team employees
15) Delete clients without health insurance and low transport importance
16) Import data from file
17) Export data to file
18) Quit
8
Enter Client SSN: 4567890123
Doctor Name: Kiran
Doctor Phone: 4748383905

```

**Output:**

No affected tables, as you can see we can retrieve doctor\_name and doctor\_phone\_no for that particular client through client\_SSN.

**6.9. Screenshots showing the testing of query 9**

```
+++
9
Enter start date (yyyy-mm-dd): 2018-04-03
Enter end date (yyyy-mm-dd): 2024-03-23
Employee SSN | Total Expenses
6124363754 | 254673.00
6119390246 | 200000.00
6281392353 | 35672.00

9
Enter start date (yyyy-mm-dd): 2014-01-20
Enter end date (yyyy-mm-dd): 2023-05-23
Employee SSN | Total Expenses
6119390246 | 200000.00
6281392353 | 35672.00

Select an option:
1) Add new team
2) Add new client and associate with teams
3) Add new volunteer and associate with teams
4) Enter volunteer hours
5) Add new employee and associate with teams
6) Enter employee expense
7) Add new donor with donations
8) Get client doctor information
9) Get total expenses by employee
10) Get volunteers for client
11) Get teams founded after a date
12) Get all people with emergency contacts
13) Get employee donors total donations
14) Increase salary for multi-team employees
15) Delete clients without health insurance and low transport importance
16) Import data from file
17) Export data to file
18) Quit
```

**Output:**

Executed and printed all the expenses which falls under particular time period.

**6.10. Screenshots showing the testing of query 10**

**10**

Enter Client SSN: **1234567890**

Volunteer SSN | Training Location

5678901347 | Univeristy

6011002344 | OU

6011003102 | dreston

Select an option:

- 1) Add new team
- 2) Add new client and associate with teams
- 3) Add new volunteer and associate with teams
- 4) Enter volunteer hours
- 5) Add new employee and associate with teams
- 6) Enter employee expense
- 7) Add new donor with donations
- 8) Get client doctor information
- 9) Get total expenses by employee
- 10) Get volunteers for client
- 11) Get teams founded after a date
- 12) Get all people with emergency contacts
- 13) Get employee donors total donations
- 14) Increase salary for multi-team employees
- 15) Delete clients without health insurance and low transport importance
- 16) Import data from file
- 17) Export data to file
- 18) Quit

**10**

Enter Client SSN: **4567890123**

Volunteer SSN | Training Location

5678901347 | Univeristy

6011002409 | campus

6011002456 | redstron

6011002456 | redstron

Retrieved successfully volunteer list that support particular client.

#### 6.11. Screenshots showing the testing of query 11

**11**

Enter date (yyyy-mm-dd): **2014-03-12**

Teams founded after 2014-03-12:

Team Name: E-Team

Team Name: D-Team

Team Name: C-Team

Team Name: A-team

Team Name: B-Team

Select an option:

- 1) Add new team
- 2) Add new client and associate with teams
- 3) Add new volunteer and associate with teams
- 4) Enter volunteer hours
- 5) Add new employee and associate with teams
- 6) Enter employee expense
- 7) Add new donor with donations
- 8) Get client doctor information
- 9) Get total expenses by employee
- 10) Get volunteers for client
- 11) Get teams founded after a date
- 12) Get all people with emergency contacts
- 13) Get employee donors total donations
- 14) Increase salary for multi-team employees
- 15) Delete clients without health insurance and low transport importance
- 16) Import data from file
- 17) Export data to file
- 18) Quit

**11**

Enter date (yyyy-mm-dd): **2022-01-01**

Teams founded after 2022-01-01:

Team Name: C-Team

Team Name: A-team

Team Name: B-Team

**Output:**

Retrieved successfully the teams founder after particular date.

#### **6.12. Screenshots showing the testing of query 12**

Since I did not able to insert any values in the emergency contact, need to insert dummy values for the all the SSN's. The dummy data is as follows:

```

732  INSERT INTO Emergency_contact (SSN, phone_no, name, relation) VALUES
733  ('1234567890', '1234567890', 'Alice', 'Friend'),
734  ('2345678901', '2345678901', 'Bob', 'Parent'),
735  ('3456789012', '3456789012', 'Charlie', 'Sibling'),
736  ('4567890123', '4567890123', 'Diana', 'Spouse'),
737  ('5678901234', '5678901234', 'Eva', 'Friend'),
738  ('5678901347', '5678901347', 'Frank', 'Parent'),
739  ('6011002344', '6011002344', 'Grace', 'Sibling'),
740  ('6011002409', '6011002409', 'Henry', 'Parent'),
741  ('6011002456', '6011002456', 'Isabella', 'Friend'),
742  ('6011003102', '6011003102', 'John', 'Spouse'),
743  ('6011003203', '6011003203', 'Kathy', 'Friend'),
744  ('6011014562', '6011014562', 'Leo', 'Sibling'),
745  ('6119390246', '6119390246', 'Mia', 'Parent'),
746  ('6124363754', '6124363754', 'Nina', 'Friend'),
747  ('6281392353', '6281392353', 'Olivia', 'Spouse'),
748  ('6281392467', '6281392467', 'Paul', 'Parent'),
749  ('6281456789', '6281456789', 'Quinn', 'Sibling'),
750  ('6381557890', '6381557890', 'Rachel', 'Friend'),
751  ('6437882899', '6437882899', 'Steve', 'Parent'),
752  ('6583938838', '6583938838', 'Tina', 'Sibling');

```

## Output

12	SSN	Name	Main Phone	Email	Emergency Phone	Emergency Contact Name	Relation
	1234567890	Feroz	4056783245	feroz@gmail.com	1234567890	Alice	Friend
	2345678901	Jackson	4563217894	jackson123@gmail.com	2345678901	Bob	Parent
	3456789012	Nigel	5673289458	nigel324@gmail.com	3456789012	Charlie	Sibling
	4567890123	Darin	5467237922	darin452@gmail.com	4567890123	Diana	Spouse
	5678901234	Indira	4748304093	indira54@gmail.com	5678901234	Eva	Friend
	5678901347	Adam	4567829302	adamdi@gmail.com	5678901347	Frank	Parent
	6011002344	Lyndi	45672840234	lyndi12@gmail.com	6011002344	Grace	Sibling
	6011002409	Sara	5637848029	sara@gmail.com	6011002409	Henry	Parent
	6011002456	Cornner	5638932757	corner@gmail.com	6011002456	Isabella	Friend
	6011003102	Adam	4638749203	adamc@gmail.com	6011003102	John	Spouse
	6011003203	Dexon	5623748038	dexon@gmail.com	6011003203	Kathy	Friend
	6011014562	Roshini	4705995025	roshini@gmail.com	6011014562	Leo	Sibling
	6119390246	Deepika	5723980278	deepika@gmail.com	6119390246	Mia	Parent
	6124363754	Reena	3528639278	reena24@gmail.com	6124363754	Nina	Friend
	6281392353	Ava	5768373833	avak@gmail.com	6281392353	Olivia	Spouse
	6281392467	Jai	5749803229	jai2@gmail.com	6281392467	Paul	Parent
	6281456789	Srikanth	4739238297	sri@gmail.com	6281456789	Quinn	Sibling
	6381557890	Julie	4684739839	juile@gmail.com	6381557890	Rachel	Friend
	6437882899	Sreeja	5647308409	sreeja@gmail.com	6437882899	Steve	Parent
	6583938838	Teja	6483984038	teja@gmail.com	6583938838	Tina	Sibling

## 6.13. Screenshots showing the testing of query 13

```

10) quit
13
Name | SSN | Total Donations | Anonymous Donor
Ava | 6281392353 | 453123.00 | Yes

```

Retrieved the donor who is also employee

## 6.14. Screenshots showing the testing of query 14

Before running the query Expense and reports table

```

500    select *from Employee;
501    select *from Reports;

```

**Results**    **Messages**

	SSN	emp_sal	marital_status	hire_date
1	6011003203	456783.00	single	2023-04-12
2	6011014562	678904.00	single	2022-04-12
3	6119390246	6484498.00	single	2012-04-03
4	6124363754	363879.00	single	2014-04-23
5	6281392353	45367.00	single	2011-03-23

	team_name	Employee_SSN	report_date	description
1	A-Team	6011003203	2023-05-13	description1
2	B-Team	6011003203	2023-05-13	description1
3	C-Team	6119390246	2015-04-02	description
4	D-Team	6011014562	2023-05-13	description2
5	E-Team	6119390246	2015-04-02	description

After running the query 14, the employees who have more than 1 team assigned has salary increased by 10%. As you observe 1<sup>st</sup> row and 3<sup>rd</sup> have more than 1 team so salary increased by 10%.

	SSN	emp_sal	marital_status	hire_date
1	6011003203	502461.30	single	2023-04-12
2	6011014562	678904.00	single	2022-04-12
3	6119390246	7132947.80	single	2012-04-03
4	6124363754	363879.00	single	2014-04-23
5	6281392353	45367.00	single	2011-03-23
	team_name	Employee_SSN	report_date	description
1	A-Team	6011003203	2023-05-13	description1
2	B-Team	6011003203	2023-05-13	description1
3	C-Team	6119390246	2015-04-02	description
4	D-Team	6011014562	2023-05-13	description2
5	E-Team	6119390246	2015-04-02	description

### 6.15. Screenshots showing the testing of query 15

Below are tables (Client, Insurance\_policy,needs and works)

	SSN	doctor_name	doctor_phone_no	date_assigned
1	1234567890	Tyson	4537892390	2022-06-09
2	2345678901	Nikhil	6543278904	2023-03-04
3	3456789012	Magaratte	6547893456	2019-05-22
4	4567890123	Kiran	4748383905	2010-08-11
5	5678901234	Nag	4638439740	2018-09-09

	policy_id	Client_SSN	name	address	type
1	1	1234567890	Life Insurance	123 Main St	life
2	2	2345678901	Auto Insurance	456 Oak St	auto
3	3	3456789012	Home Insurance	789 Pine St	home
4	4	4567890123	Health Insurance	101 Maple St	health

	Client_SSN	needs_type	importance_value
1	1234567890	transportation	3
2	2345678901	transportation	6
3	3456789012	transportation	4
4	4567890123	transportation	8

	team_name	Client_SSN	active_inactive
1	A-Team	1234567890	0
2	A-Team	5678901234	1
3	B-Team	3456789012	0
4	B-Team	4567890123	1
5	C-Team	1234567890	0
6	D-Team	2345678901	1
7	D-Team	4567890123	1
8	E-Team	4567890123	1

After running the query

Results    Messages

	SSN	doctor_name	doctor_phone_no	date_assigned
1	2345678901	Nikhil	6543278904	2023-03-04
2	4567890123	Kiran	4748383905	2010-08-11

	policy_id	Client_SSN	name	address	type
1	2	2345678901	Auto Insurance	456 Oak St	auto
2	4	4567890123	Health Insurance	101 Maple St	health

	Client_SSN	needs_type	importance_value
1	2345678901	transportation	6
2	4567890123	transportation	8

	team_name	Client_SSN	active_inactive
1	B-Team	4567890123	1
2	D-Team	2345678901	1
3	D-Team	4567890123	1
4	E-Team	4567890123	1

All client tables are deleted who do not have health insurance and whose value of importance for transportation is less than 5

#### 6.16. Screenshots showing the testing of the import and export options

Import

Input file

The screenshot shows a Java application window with a title bar 'team:' and a menu bar with 'File', 'Edit', and 'View' options. Below the menu is a list of team entries:

```
F-team,sports,2019-03-12
G-team,debat,2023-03-04
H-team,writing,2024-03-21
```

### Java execution

```
18) Quit
16
Enter the input file name: teams.txt
Team added: F-team
Team added: G-team
Team added: H-team
Import completed.
```

### Output:

---

Results    Messages

	team_name	team_type	team_date
1	A-team	sports	2023-01-13
2	B-Team	swimming	2024-12-23
3	C-Team	singing	2022-03-12
4	D-Team	dancing	2021-03-13
5	E-Team	Debat	2019-06-24
6	F-team	sports	2019-03-12
7	G-team	debat	2023-03-04
8	H-team	writing	2024-03-21

### Export

### PAN Table

	SSN	name	gender	profession	mailing_address	email_address	phone_no	mailing_list
1	1234567890	Feroz	M	Software	123 Eastern Avenue	feroz@gmail.com	4056783245	1
2	2345678901	Jackson	M	Singer	234 Preston Woods	jackson123@gmail.com	4563217894	1
3	3456789012	Nigel	M	Teacher	214 parkwood street	nigel324@gmail.com	5673289458	1
4	4567890123	Darin	M	Dancer	220 Deonne Circle	darin452@gmail.com	5467237922	0
5	5678901234	Indira	F	Singer	532 eastern street	indira54@gmail.com	4748304093	1
6	5678901347	Adam	M	Student	129	adamdi@gmail.com	4567289302	1
7	6011002344	Lyndi	F	Student	145	lyndi12@gmail.com	45672840234	1
8	6011002409	Sara	F	chef	456	sara@gmail.com	5637848029	0
9	6011002456	Cornner	M	Dancer	356	corner@gmail.com	5638932757	0
10	6011003102	Adam	M	Singer	3727	adamc@gmail.com	4638749203	1
11	6011003203	Dexon	F	Painter	2356	dexon@gmail.com	5623748038	1
12	6011014562	Roshini	F	Teacher	931	roshini@gmail.com	4705995025	1
13	6119390246	Deepika	F	Scientist	3452	deepika@gmail.com	5723980278	0
14	6124363754	Reena	f	flightattendant	3432	reena24@gmail.com	3528639278	1
15	6281392353	Ava	F	rider	5376	avak@gmail.com	5768373833	0
16	6281392467	Jai	M	Cricketer	123 weston street	jai2@gmail.com	5749803229	1
17	6281456789	Srikanth	M	Chef	123 classen blvd	sri@gmail.com	4739238297	1
18	6381557890	Julie	F	Doctor	463 western lake	juile@gmail.com	4684739839	0
19	6437882899	Sreeja	F	writer	939 deonne circle	sreeja@gmail.com	5647308409	1
20	6583938838	Teja	M	Engineer	2200 classen blvd	teja@gmail.com	6483984038	1

## Java Execution

```
~~~ 
17
Enter the output file name: output.txt
Mailing list exported to output.txt
~~~~~
```

## Exported File:

.settings	11/13/2024 6:42 PM	File folder
bin	11/13/2024 6:45 PM	File folder
src	11/14/2024 2:25 AM	File folder
.classpath	11/13/2024 6:45 PM	CLASSPATH File
.project	11/13/2024 6:42 PM	PROJECT File
data_export	11/14/2024 2:13 AM	Microsoft Excel Com...
export	11/14/2024 2:13 AM	File
output	11/14/2024 3:29 PM	Text Document
teams	11/14/2024 3:25 PM	Text Document

Name, Mailing Address  
Feroz, 123 Eastern Avenue  
Jackson, 234 Preston Woods  
Nigel, 214 parkwood street  
Indira, 532 eastern street  
Adam, 129  
Lyndi, 145  
Darin, 3727  
Dexon, 2356  
Roshini, 931  
Reena, 3432  
Jai, 123 weston street  
Srikanth, 123 classen blvd  
Sreeja, 939 deonne circle  
Teja, 2200 classen blvd

## 6.17. Screenshots showing the testing of three types of errors

### Primary Key Constraint:

Tried to insert team with same name

```
1 Enter team name: A-Team
Enter team type: sports
Enter team date (yyyy-mm-dd): 2034-02-12
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: Violation of PRIMARY KEY constraint 'PK_Teams_29E35E0D44188A22'. Cannot insert duplicate key in object 'dbo.Teams'. The duplicate
   at com.microsoft.sqlserver.jdbc@12.8.1.jre11/com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDatabaseError(SQLServerException.java:270)
   at com.microsoft.sqlserver.jdbc@12.8.1.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.getNextResult(SQLServerStatement.java:1735)
   at com.microsoft.sqlserver.jdbc@12.8.1.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.doExecutePreparedStatement(SQLServerPreparedStatement.java:675)
   at com.microsoft.sqlserver.jdbc@12.8.1.jre11/com.microsoft.sqlserver.jdbc.TDSCommand.execute(IOBuffer.java:7745)
   at com.microsoft.sqlserver.jdbc@12.8.1.jre11/com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCommand(SQLServerConnection.java:4391)
   at com.microsoft.sqlserver.jdbc@12.8.1.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeCommand(SQLServerStatement.java:276)
   at com.microsoft.sqlserver.jdbc@12.8.1.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeStatement(SQLServerStatement.java:246)
   at com.microsoft.sqlserver.jdbc@12.8.1.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.execute(SQLServerPreparedStatement.java:571)
   at Individual_project.addNewTeam(Individual_project.java:98)
```

### Input Mismatch Error

```
6 Enter Employee SSN: ffhfk
Enter expense description: hfsjd
Enter expense date (yyyy-mm-dd): 372893
Enter expense amount: jdf
Exception in thread "main" java.util.InputMismatchException
   at java.base/java.util.Scanner.throwFor(Scanner.java:964)
   at java.base/java.util.Scanner.next(Scanner.java:1619)
   at java.base/java.util.Scanner.nextDouble(Scanner.java:2590)
   at Individual_project.enterEmployeeExpense(Individual_project.java:315)
   at Individual_project.main(Individual project.java:35)
```

Tried to insert same number of hours for two months. As we have a constraint the volunteers do not work the same number of hours each month. The system is throwing an error

```
18) Quit
4
Enter team name: A-Team
Enter volunteer SSN: 5678901347
Enter hours worked: 23
Enter month (Jan, Feb, etc.): Jan
Exception in thread "main" com.microsoft.sqlserver.jdbc.SQLServerException: This volunteer already logged 23 hours for the team A-Team in a previous month. Please enter a different number of hours.
   at com.microsoft.sqlserver.jdbc@12.8.1.jre11/com.microsoft.sqlserver.jdbc.SQLServerException.makeFromDatabaseError(SQLServerException.java:270)
   at com.microsoft.sqlserver.jdbc@12.8.1.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.getNextResult(SQLServerStatement.java:1735)
   at com.microsoft.sqlserver.jdbc@12.8.1.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.doExecutePreparedStatement(SQLServerPreparedStatement.java:675)
   at com.microsoft.sqlserver.jdbc@12.8.1.jre11/com.microsoft.sqlserver.jdbc.TDSCommand.execute(IOBuffer.java:7745)
   at com.microsoft.sqlserver.jdbc@12.8.1.jre11/com.microsoft.sqlserver.jdbc.SQLServerConnection.executeCommand(SQLServerConnection.java:4391)
   at com.microsoft.sqlserver.jdbc@12.8.1.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeCommand(SQLServerStatement.java:276)
   at com.microsoft.sqlserver.jdbc@12.8.1.jre11/com.microsoft.sqlserver.jdbc.SQLServerStatement.executeStatement(SQLServerStatement.java:246)
   at com.microsoft.sqlserver.jdbc@12.8.1.jre11/com.microsoft.sqlserver.jdbc.SQLServerPreparedStatement.execute(SQLServerPreparedStatement.java:571)
   at Individual_project.enterVolunteerHours(Individual_project.java:233)
   at Individual_project.main(Individual project.java:33)
```

## 6.18. Screenshots showing the testing of the quit option

```
Select an option:
1) Add new team
2) Add new client and associate with teams
3) Add new volunteer and associate with teams
4) Enter volunteer hours
5) Add new employee and associate with teams
6) Enter employee expense
7) Add new donor with donations
8) Get client doctor information
9) Get total expenses by employee
10) Get volunteers for client
11) Get teams founded after a date
12) Get all people with emergency contacts
13) Get employee donors total donations
14) Increase salary for multi-team employees
15) Delete clients without health insurance and low transport importance
16) Import data from file
17) Export data to file
18) Quit
18
Exiting program.
```