



A CLIENT & DONOR DATABASE SYSTEM

EMILY C. BLACK, EMILY@OU.EDU, 112945795



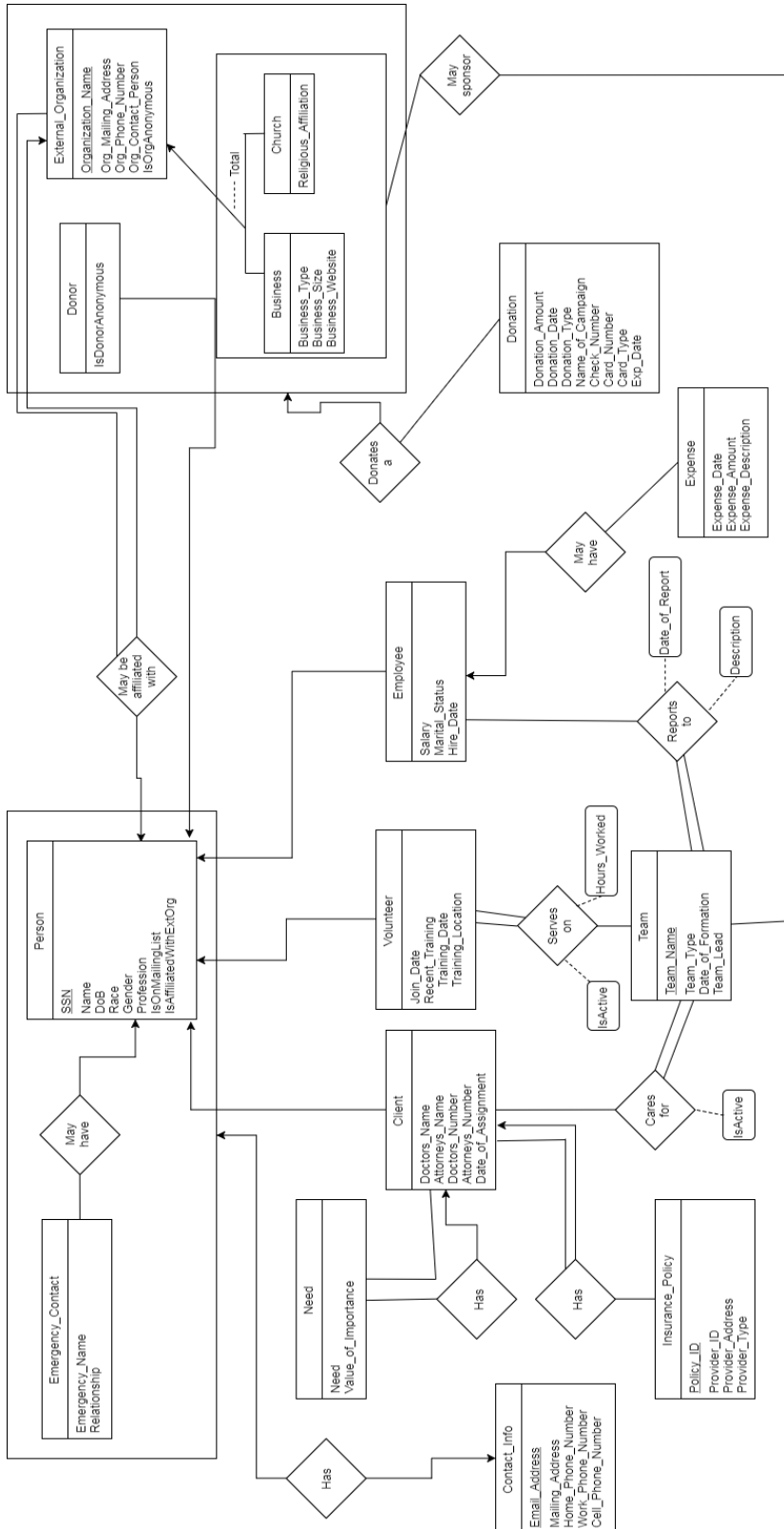
FALL 2017 – CS 4513
Taught by Dr. Le Gruenwald

Tasks Performed

| | |
|--|----|
| Task 1. | 3 |
| 1.1. ER Diagram..... | 3 |
| 1.2. Relational Database Schema..... | 4 |
| Task 2. Data Dictionary | 5 |
| Task 3. | 8 |
| 3.1. Discussion of storage structures for tables..... | 8 |
| 3.2. Discussion of storage structures for tables (Oracle 12c) | 10 |
| Task 4. SQL and text files showing the creation of tables in Oracle 12c..... | 11 |
| Task 5. Script file showing the entire Java program and its successful compilation | 17 |
| PL/SQL Queries | 17 |
| Java Program | 31 |
| Task 6. Java program Execution..... | 60 |
| 6.1. Script file showing the testing of query 1 | 60 |
| 6.2. Script file showing the testing of query 2 | 61 |
| 6.3. Script file showing the testing of query 3 | 64 |
| 6.4. Script file showing the testing of query 4 | 67 |
| 6.5. Script file showing the testing of query 5 | 69 |
| 6.6. Script file showing the testing of query 6 | 73 |
| 6.7. Script file showing the testing of query 7 | 74 |
| 6.8. Script file showing the testing of query 8 | 76 |
| 6.9. Script file showing the testing of query 9 | 79 |
| 6.10. Script file showing the testing of query 10 | 81 |
| 6.11. Script file showing the testing of query 11 | 82 |
| 6.12. Script file showing the testing of query 12 | 83 |
| 6.13. Script file showing the testing of query 13 | 83 |
| 6.14. Script file showing the testing of query 14 | 84 |
| 6.15. Script file showing the testing of query 15 | 85 |
| 6.16. Script file showing the testing of query 16 | 85 |
| 6.17. Script file showing the testing of query 17 | 87 |
| 6.18. Script file showing the testing of the import and export options | 89 |
| 6.19. Script file showing the testing of three types of errors | 94 |
| 6.20. Script file showing the testing of the quit option | 96 |

Task 1.

1.1. ER Diagram



1.2. Relational Database Schema

| Primary Key, Foreign Key, Table Name |
|--------------------------------------|
|--------------------------------------|

Contact_Info (Email_Address, Mailing_Address, Home_Phone_Number, Work_Phone_Number, Cell_Phone_Number)

Person (SSN, Email_Address, Name, DoB, Race, Gender, Profession, IsOnMailingList, IsAffiliatedWithExtOrg)

Emergency_Contact (SSN, Email_Address, Emergency_Name, Relationship)

External_Organization (Organization_Name, SSN, Org_Mailing_Address, Org_Phone_Number, Org_Contact_Person, IsOrgAnonymous)

Donor (SSN, IsDonorAnonymous)

Business (Organization_Name, Business_Type, Business_Size, Business_Website)

Church (Organization_Name, Religious_Affiliation)

Team (Team_Name, Team_Type, Date_of_Formation, Team_Lead)

Reports_to (Team_Name, SSN, Date_of_Report, Description)

May_Sponsor (Organization_Name, Team_Name)

Donation (SSN, Organization_Name, Donation_Amount, Donation_Date, Donation_Type, Name_of_Campaign, Check_Number, Card_Number, Card_Type, Exp_Date)

Employee (SSN, Salary, Martial_Status, Hire_Date)

Expense (SSN, Expense_Date, Expense_Amount, Expense_Description)

Volunteer (SSN, Join_Date, Training_Date, Training_Location)

Serves_on (SSN, Team_Name, IsActive, Jan_Hours_Worked, Feb_Hours_Worked, March_Hours_Worked, April_Hours_Worked, May_Hours_Worked, June_Hours_Worked, July_Hours_Worked, Aug_Hours_Worked, Sept_Hours_Worked, Oct_Hours_Worked, Nov_Hours_Worked, Dec_Hours_Worked)

Client (SSN, Doctors_Name, Attorneys_Name, Doctors_Number, Attorneys_Number, Date_of_Assignment)

Cares_for (SSN, Team_Name, IsActive)

Insurance_Policy (Policy_ID, SSN, Provider_ID, Provider_Address, Provider_Type)

Need (SSN, Need, Value_of_Importance)

Task 2. Data Dictionary

PK = PRIMARY KEY, FK = FOREIGN KEY

| TABLE NAME | COLUMN NAME | DATA TYPE | DATA SIZE (BYTES) | CONSTRAINT TYPE/TABLE |
|-------------------|-----------------------|-----------|-------------------|-----------------------|
| BUSINESS | ORGANIZATION_NAME | VARCHAR2 | 255 | FK/EXT._ORG. |
| BUSINESS | BUSINESS_TYPE | VARCHAR2 | 255 | --- |
| BUSINESS | BUSINESS_SIZE | NUMBER | 38 | --- |
| BUSINESS | BUSINESS_WEBSITE | VARCHAR2 | 255 | --- |
| CARES_FOR | SSN | NUMBER | 38 | FK/PERSON |
| CARES_FOR | TEAM_NAME | VARCHAR2 | 255 | FK/TEAM |
| CARES_FOR | ISACTIVE | VARCHAR2 | 1 | --- |
| CHURCH | ORGANIZATION_NAME | VARCHAR2 | 255 | FK/EXT._ORG. |
| CHURCH | RELIGIOUS_AFFILIATION | VARCHAR2 | 255 | --- |
| CLIENT | SSN | NUMBER | 38 | FK/PERSON |
| CLIENT | DOCTORS_NAME | VARCHAR2 | 255 | --- |
| CLIENT | ATTORNEYS_NAME | VARCHAR2 | 255 | --- |
| CLIENT | DOCTORS_NUMBER | NUMBER | 38 | --- |
| CLIENT | ATTORNEYS_NUMBER | NUMBER | 38 | --- |
| CLIENT | DATE_OF_ASSIGNMENT | NUMBER | 6 | --- |
| CONTACT_INFO | EMAIL_ADDRESS | VARCHAR2 | 255 | PK |
| CONTACT_INFO | MAILING_ADDRESS | VARCHAR2 | 255 | --- |
| CONTACT_INFO | HOME_PHONE_NUMBER | NUMBER | 38 | --- |
| CONTACT_INFO | WORK_PHONE_NUMBER | NUMBER | 38 | --- |
| CONTACT_INFO | CELL_PHONE_NUMBER | NUMBER | 38 | --- |
| DONATION | SSN | NUMBER | 38 | FK/PERSON |
| DONATION | ORGANIZATION_NAME | VARCHAR2 | 255 | FK/EXT._ORG. |
| DONATION | DONATION_AMOUNT | NUMBER | 38 | -- |
| DONATION | DONATION_TYPE | VARCHAR2 | 255 | --- |
| DONATION | NAME_OF_CAMPAIGN | VARCHAR2 | 255 | --- |
| DONATION | CHECK_NUMBER | NUMBER | 38 | --- |
| DONATION | CARD_NUMBER | NUMBER | 38 | --- |
| DONATION | CARD_TYPE | VARCHAR | 255 | --- |
| DONATION | EXP_DATE | NUMBER | 4 | --- |
| DONOR | SSN | NUMBER | 38 | FK/PERSON |
| DONOR | ISDONORANONYMOUS | VARCHAR2 | 1 | --- |
| EMERGENCY_CONTACT | SSN | NUMBER | 38 | FK/PERSON |
| EMERGENCY_CONTACT | EMAIL_ADDRESS | VARCHAR2 | 255 | FK/CON._INFO |
| EMERGENCY_CONTACT | EMERGENCY_NAME | VARCHAR2 | 255 | --- |
| EMERGENCY_CONTACT | RELATIONSHIP | VARCHAR2 | 255 | --- |
| EMPLOYEE | SSN | NUMBER | 38 | FK/PERSON |
| EMPLOYEE | SALARY | NUMBER | 38 | --- |
| EMPLOYEE | MARTIAL_STATUS | VARCHAR2 | 255 | --- |
| EMPLOYEE | HIRE_DATE | NUMBER | 6 | --- |
| EXPENSE | SSN | NUMBER | 38 | FK/PERSON |
| EXPENSE | EXPENSE_DATE | NUMBER | 6 | --- |

| | | | | |
|-----------------------|------------------------|----------|-----|--------------|
| EXPENSE | EXPENSE_AMOUNT | NUMBER | 38 | --- |
| EXPENSE | EXPENSE_DESCRIPTION | VARCHAR2 | 255 | --- |
| EXTERNAL_ORGANIZATION | ORGANIZATION_NAME | VARCHAR2 | 255 | PK |
| EXTERNAL_ORGANIZATION | SSN | NUMBER | 38 | FK/PERSON |
| EXTERNAL_ORGANIZATION | ORG_MAILING_ADDRESS | VARCHAR2 | 255 | --- |
| EXTERNAL_ORGANIZATION | ORG_PHONE_NUMBER | NUMBER | 38 | --- |
| EXTERNAL_ORGANIZATION | ORG_CONTACT_PERSON | VARCHAR2 | 255 | --- |
| EXTERNAL_ORGANIZATION | ISORGANONYMOUS | VARCHAR2 | 1 | --- |
| INSURANCE_POLICY | POLICY_ID | NUMBER | 38 | PK |
| INSURANCE_POLICY | SSN | NUMBER | 38 | FK/PERSON |
| INSURANCE_POLICY | PROVIDER_ID | NUMBER | 38 | --- |
| INSURANCE_POLICY | PROVIDER_ADDRESS | VARCHAR2 | 255 | --- |
| INSURANCE_POLICY | PROVIDER_TYPE | VARCHAR2 | 255 | --- |
| MAY_SPONSOR | ORGANIZATION_NAME | VARCHAR2 | 255 | FK/EXT._ORG. |
| MAY_SPONSOR | TEAM_NAME | VARCHAR2 | 255 | FK/TEAM |
| NEED | SSN | NUMBER | 38 | FK/PERSON |
| NEED | NEED | VARCHAR2 | 255 | --- |
| NEED | VALUE_OF_IMPORTANCE | NUMBER | 2 | --- |
| PERSON | SSN | NUMBER | 38 | PK |
| PERSON | EMAIL_ADDRESS | VARCHAR2 | 255 | FK/CON._INFO |
| PERSON | NAME | VARCHAR2 | 255 | --- |
| PERSON | DOB | NUMBER | 6 | --- |
| PERSON | RACE | VARCHAR2 | 255 | --- |
| PERSON | GENDER | VARCHAR2 | 1 | --- |
| PERSON | PROFESSION | VARCHAR2 | 255 | --- |
| PERSON | ISONMAILINGLIST | VARCHAR2 | 1 | --- |
| PERSON | ISAFFILIATEDWITHEXTORG | VARCHAR2 | 1 | --- |
| SERVES_ON | SSN | NUMBER | 38 | FK/PERSON |
| SERVES_ON | TEAM_NAME | VARCHAR2 | 255 | FK/TEAM |
| SERVES_ON | ISACTIVE | VARCHAR2 | 1 | --- |
| SERVES_ON | JAN_HOURS_WORKED | NUMBER | 38 | --- |
| SERVES_ON | FEB_HOURS_WORKED | NUMBER | 38 | --- |
| SERVES_ON | MARCH_HOURS_WORKED | NUMBER | 38 | --- |
| SERVES_ON | APRIL_HOURS_WORKED | NUMBER | 38 | --- |
| SERVES_ON | MAY_HOURS_WORKED | NUMBER | 38 | --- |
| SERVES_ON | JUNE_HOURS_WORKED | NUMBER | 38 | --- |
| SERVES_ON | JULY_HOURS_WORKED | NUMBER | 38 | --- |
| SERVES_ON | AUG_HOURS_WORKED | NUMBER | 38 | --- |
| SERVES_ON | SEPT_HOURS_WORKED | NUMBER | 38 | --- |
| SERVES_ON | OCT_HOURS_WORKED | NUMBER | 38 | --- |
| SERVES_ON | NOV_HOURS_WORKED | NUMBER | 38 | --- |
| SERVES_ON | DEC_HOURS_WORKED | NUMBER | 38 | --- |
| TEAM | TEAM_NAME | VARCHAR2 | 255 | PK |
| TEAM | TEAM_TYPE | VARCHAR2 | 255 | --- |
| TEAM | DATE_OF_FORMATION | NUMBER | 6 | --- |
| TEAM | TEAM_LEAD | VARCHAR2 | 255 | --- |

| | | | | |
|------------|-------------------|----------|-----|-----------|
| REPORTS_TO | TEAM_NAME | VARCHAR2 | 255 | FK/TEAM |
| REPORTS_TO | SSN | NUMBER | 38 | FK/PERSON |
| REPORTS_TO | DATE_OF_REPORT | NUMBER | 6 | --- |
| REPORTS_TO | DESCRIPTION | VARCHAR2 | 255 | --- |
| VOLUNTEER | SSN | NUMBER | 38 | FK/PERSON |
| VOLUNTEER | JOIN_DATE | NUMBER | 6 | --- |
| VOLUNTEER | TRAINING_DATE | NUMBER | 6 | --- |
| VOLUNTEER | TRAINING_LOCATION | VARCHAR2 | 255 | --- |

Task 3.

3.1. Discussion of storage structures for tables

So as far as storage structures go, I broke mine down into three different categories: **heap file, index sequential file and dynamic hashing**.

I used **heap files** when a table had a bunch of insertion queries.

I used **index sequential files** when a table had a bunch of searching queries.

I used **dynamic hashing** when a table had a balanced mix of insertion and searching queries.

| Table Name | Type of Query | Search Key | Frequency | Best Storage Structure |
|-------------------|---|---|---|-----------------------------------|
| Contact_Info | [2] Insert [3] Insert [5] Insert [13] Random Search [15] Random Search | ----- ----- ----- Email_Address Email_Address | 1/week 2/month 1/year 1/week 1/month | Index sequential file |
| Person | [2] Insert [3] Insert [5] Insert [10] Random Search [11] Random Search [12] Random Search [13] Random Search [14] Random Search [15] Random Search [16] Random Search [17] Delete | ----- ----- ----- SSN SSN SSN SSN SSN SSN SSN SSN | 1/week 2/month 1/year 1/week 1/month 4/year 1/week 1/week 1/month 1/year 4/year | Index sequential file |
| Emergency_Contact | [17] Random Search | SSN | 4/year | Index sequential file |
| Need | [17] Random Search | SSN | 4/year | Index sequential file |
| Insurance_Policy | [17] Random Search | Policy_ID | 4/year | Index sequential file |
| Client | [2] Insert [10] Random Search | ----- SSN SSN SSN | 1/week 1/week 1/week 4/year | Dynamic hashing with hash key SSN |

| | | | | |
|------------|--|---|---|--|
| | [13] Random Search [17] Delete | | | |
| Volunteer | [3] Insert [12] Random Search [15] Random Search | ----- SSN SSN | 2/month 4/year 1/month | Dynamic hashing with hash key SSN |
| Team | [1] Insert [5] Insert [12] Random Search [15] Random Search [16] Random Search | ----- ----- Team_Name Team_Name Team_Name | 1/month 1/year 4/year 1/month 1/year | Dynamic hashing with hash key Team_Name |
| Reports_to | [1] Insert [5] Insert [11] Random Search [12] Random Search [15] Random Search [14] Random Search [16] Random Search | ----- ----- SSN Team_Name Team_Name SSN Team_Name | 1/month 1/year 1/month 4/year 1/month 1/week 1/year | Index sequential file |
| Employee | [5] Insert [11] Random Search [14] Random Search [16] Random Search | ----- SSN SSN SSN | 1/year 1/month 1/week 1/year | Index sequential file |
| Expense | [6] Insert [11] Range Search | ----- Expense_Date | 1/day 1/month | Dynamic hashing with hash key Expense_Date |
| Donation | [9] Insert [10] Insert [14] Random Search | ----- ----- Donation_Amount | 1/day 1/week 1/week | Heap file |
| Business | [7] Insert [9] Insert | ----- ----- | 2/week 1/day | Heap file |

| | | | | |
|-----------------------|---|--|--|---|
| Church | [7] Insert [9] Insert | ----- ----- | 2/week 1/day | Heap file |
| External_Organization | [7] Insert [9] Insert [13] Range Search | ----- ----- Organization_Name | 2/week 1/day 1/week | Heap file |
| Donor | [7] Insert [8] Insert [14] Random Search | ----- ----- SSN | 2/week 1/day 1/week | Heap file |
| Cares_for | [2] Insert [12] Random Search [13] Random Search | ----- SSN, Team_Name SSN, Team_Name | 1/week 4/year 1/week | Index sequential file |
| Serves_on | [3] Insert [4] Insert [12] Random Search [15] Range Search | ----- ----- SSN, Team_Name SSN, Team_Name | 2/month 30/month 4/year 1/month | Dynamic hashing with hash key SSN |
| May_Sponsor | [7] Insert [13] Range Search | ----- Organization_Name, Team_Name | 2/week 1/week | Dynamic hashing with hash key Organization_Name |

3.2. Discussion of storage structures for tables (Oracle 12c)

Because many of the search keys above are primary keys, and with Oracle automatically indexing primary keys, below are the indexes that are most searched throughout the 17 queries, AND are not primary keys. I also did not include heap indexes.

```
CREATE INDEX DoctorsNameIndex
ON Client (Doctors_Name);
```

```
CREATE INDEX DoctorsNumberIndex
ON Client (Doctors_Number);
```

```
CREATE INDEX ExpenseAmountIndex
ON Expense (Expense_Amount);
```

```
CREATE INDEX NameIndex
ON Person (Name);
```

Task 4. SQL and text files showing the creation of tables in Oracle 12c

```
CREATE TABLE Contact_Info (
  Email_Address          VARCHAR(255) NOT NULL,
  Mailing_Address        VARCHAR(255),
  Home_Phone_Number      NUMBER(38),
  Work_Phone_Number      NUMBER(38),
  Cell_Phone_Number      NUMBER(38),
  PRIMARY KEY(Email_Address)
);
```

```
CREATE TABLE Person (
  SSN                    NUMBER(38) NOT NULL,
  Email_Address          VARCHAR(255),
  Name                   VARCHAR(255),
  DoB                    NUMBER(6),
  Race                   VARCHAR(255),
  Gender                 VARCHAR(1),
  Profession              VARCHAR(255),
  IsOnMailingList         VARCHAR(1),
  IsAffiliatedWithExtOrg  VARCHAR(1),
  PRIMARY KEY(SSN),
  FOREIGN KEY(Email_Address) REFERENCES Contact_Info
);
```

```
CREATE TABLE Emergency_Contact (
  SSN                    NUMBER(38),
  Email_Address          VARCHAR(255),
  Emergency_Name          VARCHAR(255),
  Relationship            VARCHAR(255),
  FOREIGN KEY(SSN) REFERENCES Person,
  FOREIGN KEY(Email_Address) REFERENCES Contact_Info
);
```

```
CREATE TABLE External_Organization (
  Organization_Name       VARCHAR(255) NOT NULL,
  SSN                     NUMBER(38),
  Org_Mailing_Address     VARCHAR(255),
  Org_Phone_Number        NUMBER(38),
  Org_Contact_Person      VARCHAR(255),
  IsOrgAnonymous          VARCHAR(1),
  PRIMARY KEY(Organization_Name),
  FOREIGN KEY(SSN) REFERENCES Person
);
```

```
CREATE TABLE Donor (
  SSN                     NUMBER(38),
  IsDonorAnonymous        VARCHAR(1),
```

```
FOREIGN KEY(SSN) REFERENCES Person
);
```

```
CREATE TABLE Business (
    Organization_Name      VARCHAR(255),
    Business_Type          VARCHAR(255),
    Business_Size          NUMBER(38),
    Business_Website       VARCHAR(255),
    FOREIGN KEY(Organization_Name) REFERENCES External_Organization
);
```

```
CREATE TABLE Church (
    Organization_Name      VARCHAR(255),
    Religious_Affiliation  VARCHAR(255),
    FOREIGN KEY(Organization_Name) REFERENCES External_Organization
);
```

```
CREATE TABLE Team (
    Team_Name              VARCHAR(255) NOT NULL,
    Team_Type              VARCHAR(255),
    Date_of_Formation      NUMBER(6),
    Team_Lead              VARCHAR(255),
    PRIMARY KEY(Team_Name)
);
```

```
CREATE TABLE Reports_to (
    Team_Name              VARCHAR(255),
    SSN                    NUMBER(38),
    Date_of_Report         NUMBER(6),
    Description             VARCHAR(255),
    FOREIGN KEY(Team_Name) REFERENCES Team,
    FOREIGN KEY(SSN) REFERENCES Person
);
```

```
CREATE TABLE May_Sponsor (
    Organization_Name      VARCHAR(255),
    Team_Name              VARCHAR(255),
    FOREIGN KEY(Organization_Name) REFERENCES External_Organization,
    FOREIGN KEY(Team_Name) REFERENCES Team
);
```

```
CREATE TABLE Donation (
    SSN                    NUMBER(38),
    Organization_Name      VARCHAR(255),
    Donation_Amount        NUMBER(38),
    Donation_Type          VARCHAR(255),
    Name_of_Campaign       VARCHAR(255),
    Check_Number           NUMBER(38),
```

```

Card_Number          NUMBER(38),
Card_Type             VARCHAR(255),
Exp_Date             NUMBER(4),
FOREIGN KEY(SSN) REFERENCES Person,
FOREIGN KEY(Organization_Name) REFERENCES External_Organization
);

```

```

CREATE TABLE Employee (
  SSN                NUMBER(38),
  Salary             NUMBER(38),
  Martial_Status     VARCHAR(255),
  Hire_Date          NUMBER(6),
  FOREIGN KEY(SSN) REFERENCES Person
);

```

```

CREATE TABLE Expense (
  SSN                NUMBER(38),
  Expense_Date       NUMBER(6),
  Expense_Amount     NUMBER(38),
  Expense_Description VARCHAR(255),
  FOREIGN KEY(SSN) REFERENCES Person
);

```

```

CREATE TABLE Volunteer (
  SSN                NUMBER(38),
  Join_Date          NUMBER(6),
  Training_Date      NUMBER(6),
  Training_Location  VARCHAR(255),
  FOREIGN KEY(SSN) REFERENCES Person
);

```

```

CREATE TABLE Serves_on (
  SSN                NUMBER(38),
  Team_Name          VARCHAR(255),
  IsActive           VARCHAR(1),
  Jan_Hours_Worked   NUMBER(38),
  Feb_Hours_Worked   NUMBER(38),
  March_Hours_Worked NUMBER(38),
  April_Hours_Worked NUMBER(38),
  May_Hours_Worked   NUMBER(38),
  June_Hours_Worked  NUMBER(38),
  July_Hours_Worked  NUMBER(38),
  Aug_Hours_Worked   NUMBER(38),
  Sept_Hours_Worked  NUMBER(38),
  Oct_Hours_Worked   NUMBER(38),
  Nov_Hours_Worked   NUMBER(38),
  Dec_Hours_Worked   NUMBER(38),
  FOREIGN KEY(SSN) REFERENCES Person,

```

```
FOREIGN KEY(Team_Name) REFERENCES Team
);
```

```
CREATE TABLE Client (
    SSN                NUMBER(38),
    Doctors_Name       VARCHAR(255),
    Attorneys_Name     VARCHAR(255),
    Doctors_Number     NUMBER(38),
    Attorneys_Number   NUMBER(38),
    Date_of_Assignment NUMBER(6),
    FOREIGN KEY(SSN) REFERENCES Person
);
```

```
CREATE TABLE Cares_for (
    SSN                NUMBER(38),
    Team_Name          VARCHAR(255),
    IsActive           VARCHAR(1),
    FOREIGN KEY(SSN) REFERENCES Person,
    FOREIGN KEY(Team_Name) REFERENCES Team
);
```

```
CREATE TABLE Insurance_Policy (
    Policy_ID          NUMBER(38) NOT NULL,
    SSN                NUMBER(38),
    Provider_ID        NUMBER(38),
    Provider_Address    VARCHAR(255),
    Provider_Type       VARCHAR(255),
    PRIMARY KEY(Policy_ID),
    FOREIGN KEY(SSN) REFERENCES Person
);
```

```
CREATE TABLE Need (
    SSN                NUMBER(38),
    Need               VARCHAR(255),
    Value_of_Importance NUMBER(2),
    FOREIGN KEY(SSN) REFERENCES Person
);
```

```
CREATE INDEX DoctorsNameIndex
ON Client (Doctors_Name);
```

```
CREATE INDEX DoctorsNumberIndex
ON Client (Doctors_Number);
```

```
CREATE INDEX ExpenseAmountIndex
ON Expense (Expense_Amount);
```

```
CREATE INDEX NameIndex
```

ON **Person** (Name);

Output:

Table CONTACT_INFO created.

Table PERSON created.

Table EMERGENCY_CONTACT created.

Table EXTERNAL_ORGANIZATION created.

Table DONOR created.

Table BUSINESS created.

Table CHURCH created.

Table TEAM created.

Table MAY_SPONSOR created.

Table DONATION created.

Table EMPLOYEE created.

Table EXPENSE created.

Table VOLUNTEER created.

Table SERVES_ON created.

Table CLIENT created.

Table CARES_FOR created.

Table INSURANCE_POLICY created.

Table NEED created.

Index DOCTORSNAMEINDEX created.

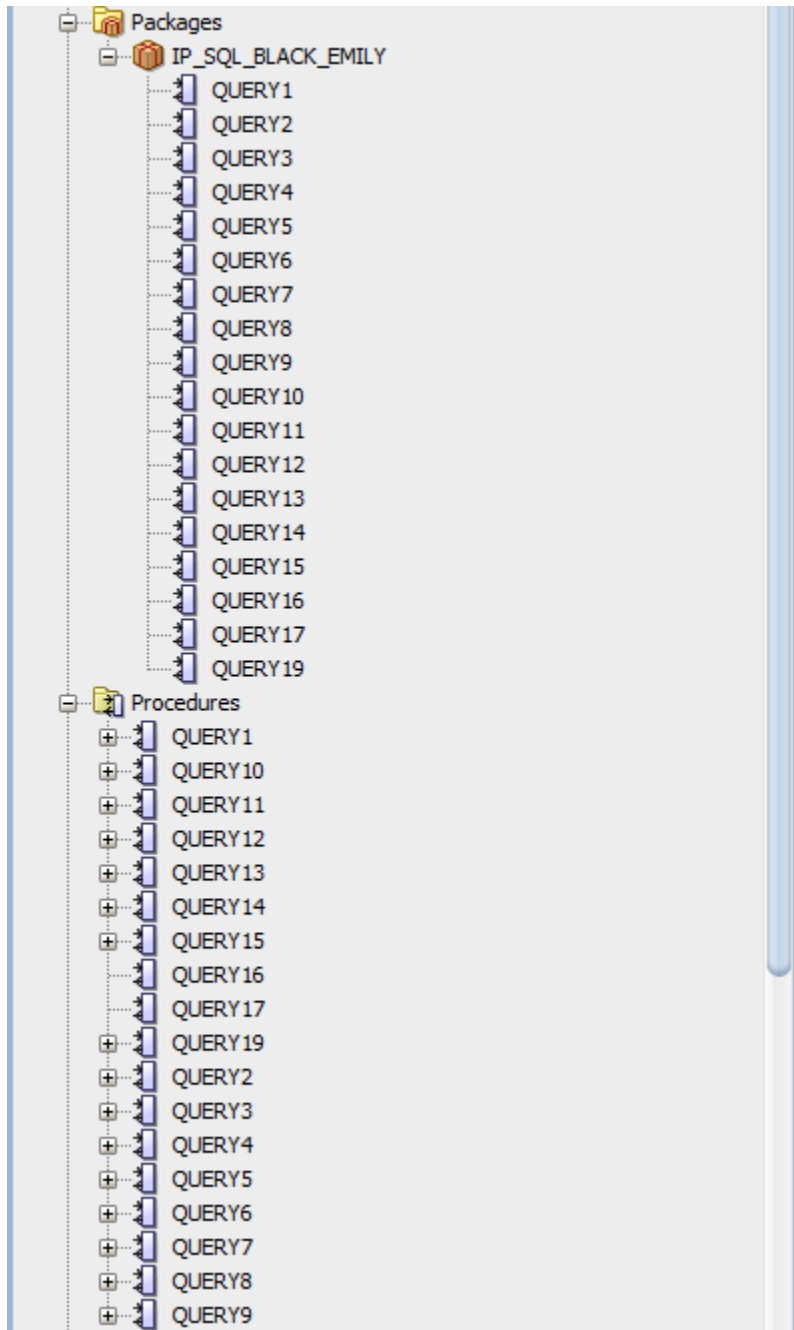
Index DOCTORSNUMBERINDEX created.

Index EXPENSEAMOUNTINDEX created.

Index NAMEINDEX created.

Task 5. Script file showing the entire Java program and its successful compilation

PL/SQL Queries



```
CREATE OR REPLACE PACKAGE IP_SQL_BLACK_EMILY AS
```

```
    PROCEDURE QUERY1
```

```
    (Team_Name_Input IN VARCHAR2, Team_Type_Input IN VARCHAR2, Date_of_Formation_Input,
    Team_Lead_Input IN VARCHAR2);
```

PROCEDURE QUERY2

(Email_Address_Input IN VARCHAR2, Mailing_Address_Input IN VARCHAR2, Home_Phone_Number_Input, Work_Phone_Number_Input, Cell_Phone_Number_Input, SSN_Input, Name_Input IN VARCHAR2, DoB_Input, Race_Input IN VARCHAR2, Gender_Input IN VARCHAR2, Profession_Input IN VARCHAR2, IsOnMailingList_Input IN VARCHAR2, IsAffiliatedWithExtOrg_Input IN VARCHAR2, Doctors_Name_Input IN VARCHAR2, Attorneys_Name_Input IN VARCHAR2, Doctors_Number_Input, Attorneys_Number_Input, Date_of_Assignment_Input, Team_Name_Input IN VARCHAR2, IsActive_Input IN VARCHAR2);

PROCEDURE QUERY3

(Email_Address_Input IN VARCHAR2, Mailing_Address_Input IN VARCHAR2, Home_Phone_Number_Input, Work_Phone_Number_Input, Cell_Phone_Number_Input, SSN_Input, Name_Input IN VARCHAR2, DoB_Input, Race_Input IN VARCHAR2, Gender_Input IN VARCHAR2, Profession_Input IN VARCHAR2, IsOnMailingList_Input IN VARCHAR2, IsAffiliatedWithExtOrg_Input IN VARCHAR2, Join_Date_Input, Training_Date_Input, Training_Location_Input IN VARCHAR2, Team_Name_Input IN VARCHAR2, IsActive_Input IN VARCHAR2, Jan_Hours_Worked_Input, Feb_Hours_Worked_Input, March_Hours_Worked_Input, April_Hours_Worked_Input, May_Hours_Worked_Input, June_Hours_Worked_Input, July_Hours_Worked_Input, Aug_Hours_Worked_Input, Sept_Hours_Worked_Input, Oct_Hours_Worked_Input, Nov_Hours_Worked_Input, Dec_Hours_Worked_Input IN NUMBER);

PROCEDURE QUERY4

(SSN_Input IN NUMBER, Jan_Hours_Worked_Input IN NUMBER, Feb_Hours_Worked_Input IN NUMBER, March_Hours_Worked_Input IN NUMBER, April_Hours_Worked_Input IN NUMBER, May_Hours_Worked_Input IN NUMBER, June_Hours_Worked_Input IN NUMBER, July_Hours_Worked_Input IN NUMBER, Aug_Hours_Worked_Input IN NUMBER, Sept_Hours_Worked_Input IN NUMBER, Oct_Hours_Worked_Input IN NUMBER, Nov_Hours_Worked_Input IN NUMBER, Dec_Hours_Worked_Input IN NUMBER);

PROCEDURE QUERY5

(Email_Address_Input IN VARCHAR2, Mailing_Address_Input IN VARCHAR2, Home_Phone_Number_Input, Work_Phone_Number_Input, Cell_Phone_Number_Input, SSN_Input, Name_Input IN VARCHAR2, DoB_Input, Race_Input IN VARCHAR2, Gender_Input IN VARCHAR2, Profession_Input IN VARCHAR2, IsOnMailingList_Input IN VARCHAR2, IsAffiliatedWithExtOrg_Input IN VARCHAR2, Salary_Input, Martial_Status_Input IN VARCHAR2, Hire_Date_Input, Team_Name_Input IN VARCHAR2, Date_of_Report_Input, Description_Input IN VARCHAR2);

PROCEDURE QUERY6

(SSN_Input, Expense_Date_Input, Expense_Amount_Input, Expense_Description_Input IN VARCHAR2);

PROCEDURE QUERY7

(Organization_Name_Input IN VARCHAR2, SSN_Input, Org_Mailing_Address_Input IN VARCHAR2, Org_Phone_Number_Input, Org_Contact_Person_Input IN VARCHAR2, IsOrgAnonymous_Input IN VARCHAR2, Team_Name_Input IN VARCHAR2);

PROCEDURE QUERY8

(Email_Address_Input IN VARCHAR2, Mailing_Address_Input IN VARCHAR2, Home_Phone_Number_Input IN NUMBER, Work_Phone_Number_Input IN NUMBER, Cell_Phone_Number_Input IN NUMBER, SSN_Input IN NUMBER, Name_Input IN VARCHAR2, DoB_Input IN NUMBER, Race_Input IN VARCHAR2, Gender_Input IN VARCHAR2, Profession_Input IN VARCHAR2, IsOnMailingList_Input IN VARCHAR2, IsAffiliatedWithExtOrg_Input IN VARCHAR2, IsDonorAnonymous_Input IN VARCHAR2);

PROCEDURE QUERY9

(Organization_Name_Input IN VARCHAR2, SSN_Input, Org_Mailing_Address_Input IN VARCHAR2, Org_Phone_Number_Input, Org_Contact_Person_Input IN VARCHAR2, IsOrgAnonymous_Input IN VARCHAR2);

PROCEDURE QUERY10

(SSN_Input, Doctors_Name_Output OUT VARCHAR2, Doctors_Number_Output OUT NUMBER);

PROCEDURE QUERY11

(Expense_Date_Input, Result_Table_Output OUT SYS_REFCURSOR);

PROCEDURE QUERY12

(SSN_Input, Result_Table_Output OUT SYS_REFCURSOR);

PROCEDURE QUERY13

(Result_Table_Output OUT SYS_REFCURSOR);

PROCEDURE QUERY14

(Donation_Result_Table_Output OUT SYS_REFCURSOR, Name_Result_Table_Output OUT SYS_REFCURSOR, Isanon_Result_Table_Output OUT SYS_REFCURSOR);

PROCEDURE QUERY15

(Result_Table_Output OUT SYS_REFCURSOR);

PROCEDURE QUERY16;

PROCEDURE QUERY17;

PROCEDURE QUERY19

(Result_Table_Output OUT SYS_REFCURSOR);

```
END IP_SQL_BLACK_EMILY;
```

```
CREATE OR REPLACE PACKAGE BODY IP_SQL_BLACK_EMILY AS
```

```
--Query 1: Enter a new team into the database
```

```
CREATE OR REPLACE PROCEDURE QUERY1
```

```
(Team_Name_Input IN VARCHAR2, Team_Type_Input IN VARCHAR2, Date_of_Formation_Input  
IN NUMBER, Team_Lead_Input IN VARCHAR2) AS
```

```
BEGIN
```

```
INSERT INTO TEAM(Team_Name, Team_Type, Date_of_Formation, Team_Lead)
```

```
VALUES(Team_Name_Input, Team_Type_Input, Date_of_Formation_Input,  
Team_Lead_Input);
```

```
END;
```

```
--Query 2: Enter a new client into the database and associate him or her with one or more teams  
(assumes the client's team already exists)
```

```
CREATE OR REPLACE PROCEDURE QUERY2
```

```
(Email_Address_Input IN VARCHAR2, Mailing_Address_Input IN VARCHAR2,  
Home_Phone_Number_Input IN NUMBER, Work_Phone_Number_Input IN NUMBER,  
Cell_Phone_Number_Input IN NUMBER, SSN_Input IN NUMBER, Name_Input IN VARCHAR2, DoB_Input  
IN NUMBER, Race_Input IN VARCHAR2, Gender_Input IN VARCHAR2, Profession_Input IN VARCHAR2,  
IsOnMailingList_Input IN VARCHAR2, IsAffiliatedWithExtOrg_Input IN VARCHAR2, Doctors_Name_Input  
IN VARCHAR2, Attorneys_Name_Input IN VARCHAR2, Doctors_Number_Input IN NUMBER,  
Attorneys_Number_Input IN NUMBER, Date_of_Assignment_Input IN NUMBER, Team_Name_Input IN  
VARCHAR2, IsActive_Input IN VARCHAR2) AS
```

```
BEGIN
```

```
INSERT INTO CONTACT_INFO(Email_Address, Mailing_Address, Home_Phone_Number,  
Work_Phone_Number, Cell_Phone_Number)
```

```
VALUES(Email_Address_Input, Mailing_Address_Input, Home_Phone_Number_Input,  
Work_Phone_Number_Input, Cell_Phone_Number_Input);
```

```
INSERT INTO PERSON(SSN, Email_Address, Name, DoB, Race, Gender, Profession,  
IsOnMailingList, IsAffiliatedWithExtOrg)
```

```
VALUES(SSN_Input, Email_Address_Input, Name_Input, DoB_Input, Race_Input,  
Gender_Input, Profession_Input, IsOnMailingList_Input, IsAffiliatedWithExtOrg_Input);
```

```
INSERT INTO CLIENT(SSN, Doctors_Name, Attorneys_Name, Doctors_Number,
Attorneys_Number, Date_of_Assignment)
```

```
VALUES(SSN_Input, Doctors_Name_Input, Attorneys_Name_Input,
Doctors_Number_Input, Attorneys_Number_Input, Date_of_Assignment_Input);
```

```
INSERT INTO CARES_FOR(SSN, Team_Name, IsActive)
```

```
VALUES(SSN_Input, Team_Name_Input, IsActive_Input);
```

```
END;
```

--Query 3: Enter a new volunteer into the database and associate him or her with one or more teams
(assumes the volunteer's team already exists)

```
CREATE OR REPLACE PROCEDURE QUERY3
```

```
(Email_Address_Input IN VARCHAR2, Mailing_Address_Input IN VARCHAR2,
Home_Phone_Number_Input IN NUMBER, Work_Phone_Number_Input IN NUMBER,
Cell_Phone_Number_Input IN NUMBER, SSN_Input IN NUMBER, Name_Input IN VARCHAR2, DoB_Input
IN NUMBER, Race_Input IN VARCHAR2, Gender_Input IN VARCHAR2, Profession_Input IN VARCHAR2,
IsOnMailingList_Input IN VARCHAR2, IsAffiliatedWithExtOrg_Input IN VARCHAR2, Join_Date_Input IN
NUMBER, Training_Date_Input IN NUMBER, Training_Location_Input IN VARCHAR2, Team_Name_Input
IN VARCHAR2, IsActive_Input IN VARCHAR2, Jan_Hours_Worked_Input IN NUMBER,
Feb_Hours_Worked_Input IN NUMBER, March_Hours_Worked_Input IN NUMBER,
April_Hours_Worked_Input IN NUMBER, May_Hours_Worked_Input IN NUMBER,
June_Hours_Worked_Input IN NUMBER, July_Hours_Worked_Input IN NUMBER,
Aug_Hours_Worked_Input IN NUMBER, Sept_Hours_Worked_Input IN NUMBER,
Oct_Hours_Worked_Input IN NUMBER, Nov_Hours_Worked_Input IN NUMBER,
Dec_Hours_Worked_Input IN NUMBER) AS
```

```
BEGIN
```

```
INSERT INTO CONTACT_INFO(Email_Address, Mailing_Address, Home_Phone_Number,
Work_Phone_Number, Cell_Phone_Number)
```

```
VALUES(Email_Address_Input, Mailing_Address_Input, Home_Phone_Number_Input,
Work_Phone_Number_Input, Cell_Phone_Number_Input);
```

```
INSERT INTO PERSON(SSN, Email_Address, Name, DoB, Race, Gender, Profession,
IsOnMailingList, IsAffiliatedWithExtOrg)
```

```
VALUES(SSN_Input, Email_Address_Input, Name_Input, DoB_Input, Race_Input,
Gender_Input, Profession_Input, IsOnMailingList_Input, IsAffiliatedWithExtOrg_Input);
```

```
INSERT INTO VOLUNTEER(SSN, Join_Date, Training_Date, Training_Location)
```

```
VALUES(SSN_Input, Join_Date_Input, Training_Date_Input, Training_Location_Input);
```

```

INSERT INTO SERVES_ON(SSN, Team_Name, IsActive, Jan_Hours_Worked,
Feb_Hours_Worked, March_Hours_Worked, April_Hours_Worked, May_Hours_Worked,
June_Hours_Worked, July_Hours_Worked, Aug_Hours_Worked, Sept_Hours_Worked,
Oct_Hours_Worked, Nov_Hours_Worked, Dec_Hours_Worked)

VALUES(SSN_Input, Team_Name_Input, IsActive_Input, Jan_Hours_Worked_Input,
Feb_Hours_Worked_Input, March_Hours_Worked_Input, April_Hours_Worked_Input,
May_Hours_Worked_Input, June_Hours_Worked_Input, July_Hours_Worked_Input,
Aug_Hours_Worked_Input, Sept_Hours_Worked_Input, Oct_Hours_Worked_Input,
Nov_Hours_Worked_Input, Dec_Hours_Worked_Input);

END;

```

--Query 4: Enter the number of hours a volunteer worked this month for a particular team (assumes volunteer already exists)

```
CREATE OR REPLACE PROCEDURE QUERY4
```

```

(SSN_Input IN NUMBER, Team_Name_Input IN VARCHAR2, IsActive_Input IN VARCHAR2,
Jan_Hours_Worked_Input IN NUMBER, Feb_Hours_Worked_Input IN NUMBER,
March_Hours_Worked_Input IN NUMBER, April_Hours_Worked_Input IN NUMBER,
May_Hours_Worked_Input IN NUMBER, June_Hours_Worked_Input IN NUMBER,
July_Hours_Worked_Input IN NUMBER, Aug_Hours_Worked_Input IN NUMBER,
Sept_Hours_Worked_Input IN NUMBER, Oct_Hours_Worked_Input IN NUMBER,
Nov_Hours_Worked_Input IN NUMBER, Dec_Hours_Worked_Input IN NUMBER) AS

```

```
BEGIN
```

```
UPDATE SERVES_ON
```

```
SET
```

```
Jan_Hours_Worked = Jan_Hours_Worked + Jan_Hours_Worked_Input,
```

```
Feb_Hours_Worked = Feb_Hours_Worked + Feb_Hours_Worked_Input,
```

```
March_Hours_Worked = March_Hours_Worked + March_Hours_Worked_Input,
```

```
April_Hours_Worked = April_Hours_Worked + April_Hours_Worked_Input,
```

```
May_Hours_Worked = May_Hours_Worked + May_Hours_Worked_Input,
```

```
June_Hours_Worked = June_Hours_Worked + June_Hours_Worked_Input,
```

```
July_Hours_Worked = July_Hours_Worked + July_Hours_Worked_Input,
```

```
Aug_Hours_Worked = Aug_Hours_Worked + Aug_Hours_Worked_Input,
```

```
Sept_Hours_Worked = Sept_Hours_Worked + Sept_Hours_Worked_Input,
```

```
Oct_Hours_Worked = Oct_Hours_Worked + Oct_Hours_Worked_Input,
```

```

Nov_Hours_Worked = Nov_Hours_Worked + Nov_Hours_Worked_Input,
Dec_Hours_Worked = Dec_Hours_Worked + Dec_Hours_Worked_Input
WHERE SSN_Input = SERVES_ON.SSN;
END;

```

--Query 5: Enter a new employee into the database and associate him or her with one or more teams (assumes the employee's team already exists)

```

CREATE OR REPLACE PROCEDURE QUERY5
(Email_Address_Input IN VARCHAR2, Mailing_Address_Input IN VARCHAR2,
Home_Phone_Number_Input IN NUMBER, Work_Phone_Number_Input IN NUMBER,
Cell_Phone_Number_Input IN NUMBER, SSN_Input IN NUMBER, Name_Input IN VARCHAR2, DoB_Input
IN NUMBER, Race_Input IN VARCHAR2, Gender_Input IN VARCHAR2, Profession_Input IN VARCHAR2,
IsOnMailingList_Input IN VARCHAR2, IsAffiliatedWithExtOrg_Input IN VARCHAR2, Salary_Input IN
NUMBER, Martial_Status_Input IN VARCHAR2, Hire_Date_Input IN NUMBER, Team_Name_Input IN
VARCHAR2, Date_of_Report_Input IN NUMBER, Description_Input IN VARCHAR2) AS
BEGIN
    INSERT INTO CONTACT_INFO(Email_Address, Mailing_Address, Home_Phone_Number,
Work_Phone_Number, Cell_Phone_Number)
    VALUES(Email_Address_Input, Mailing_Address_Input, Home_Phone_Number_Input,
Work_Phone_Number_Input, Cell_Phone_Number_Input);
    INSERT INTO PERSON(SSN, Email_Address, Name, DoB, Race, Gender, Profession,
IsOnMailingList, IsAffiliatedWithExtOrg)
    VALUES(SSN_Input, Email_Address_Input, Name_Input, DoB_Input, Race_Input,
Gender_Input, Profession_Input, IsOnMailingList_Input, IsAffiliatedWithExtOrg_Input);
    INSERT INTO EMPLOYEE(SSN, Salary, Martial_Status, Hire_Date)
    VALUES(SSN_Input, Salary_Input, Martial_Status_Input, Hire_Date_Input);
    INSERT INTO REPORTS_TO(Team_Name, SSN, Date_of_Report, Description)
    VALUES(Team_Name_Input, SSN_Input, Date_of_Report_Input, Description_Input);
END;

```

--Query 6: Enter an expense charged by an employee (assumes the employee already exists)

```

CREATE OR REPLACE PROCEDURE QUERY6

```



```
(SSN_Input IN NUMBER, Expense_Date_Input IN NUMBER, Expense_Amount_Input IN NUMBER,
Expense_Description_Input IN VARCHAR2) AS
```

```
BEGIN
```

```
INSERT INTO EXPENSE(SSN, Expense_Date, Expense_Amount, Expense_Description)
```

```
VALUES(SSN_Input, Expense_Date_Input, Expense_Amount_Input,
Expense_Description_Input);
```

```
END;
```

--Query 7: Enter a new organization and associate it to one or more PAN teams (assumes the team/employee, associated with the ext_org, already exists)

-- (EXTERNAL_ORGANIZATION => SSN may be null)

```
PROCEDURE QUERY7
```

```
(Organization_Name_Input IN VARCHAR2, SSN_Input IN NUMBER, Org_Mailing_Address_Input
IN VARCHAR2, Org_Phone_Number_Input IN NUMBER, Org_Contact_Person_Input IN VARCHAR2,
IsOrgAnonymous_Input IN VARCHAR2, Team_Name_Input IN VARCHAR2) AS
```

```
BEGIN
```

```
INSERT INTO EXTERNAL_ORGANIZATION(Organization_Name, SSN,
Org_Mailing_Address, Org_Phone_Number, Org_Contact_Person, IsOrgAnonymous)
```

```
VALUES(Organization_Name_Input, SSN_Input, Org_Mailing_Address_Input,
Org_Phone_Number_Input, Org_Contact_Person_Input, IsOrgAnonymous_Input);
```

```
INSERT INTO MAY_SPONSOR(Organization_Name, Team_Name)
```

```
VALUES(Organization_Name_Input, Team_Name_Input);
```

```
END;
```

--Query 8: Enter a new donor and associate him or her with several donations (assumes the donor's donations already exist)

```
PROCEDURE QUERY8
```

```
(Email_Address_Input IN VARCHAR2, Mailing_Address_Input IN VARCHAR2,
Home_Phone_Number_Input IN NUMBER, Work_Phone_Number_Input IN NUMBER,
Cell_Phone_Number_Input IN NUMBER, SSN_Input IN NUMBER, Name_Input IN VARCHAR2, DoB_Input
IN NUMBER, Race_Input IN VARCHAR2, Gender_Input IN VARCHAR2, Profession_Input IN VARCHAR2,
IsOnMailingList_Input IN VARCHAR2, IsAffiliatedWithExtOrg_Input IN VARCHAR2,
IsDonorAnonymous_Input IN VARCHAR2) AS
```

```

BEGIN

    INSERT INTO CONTACT_INFO(Email_Address, Mailing_Address, Home_Phone_Number,
Work_Phone_Number, Cell_Phone_Number)

        VALUES(Email_Address_Input, Mailing_Address_Input, Home_Phone_Number_Input,
Work_Phone_Number_Input, Cell_Phone_Number_Input);

    INSERT INTO PERSON(SSN, Email_Address, Name, DoB, Race, Gender, Profession,
IsOnMailingList, IsAffiliatedWithExtOrg)

        VALUES(SSN_Input, Email_Address_Input, Name_Input, DoB_Input, Race_Input,
Gender_Input, Profession_Input, IsOnMailingList_Input, IsAffiliatedWithExtOrg_Input);

    INSERT INTO DONOR(SSN, IsDonorAnonymous)

        VALUES(SSN_Input, IsDonorAnonymous_Input);

END;

```

--Query 9: Enter a new organization and associate it with several donations (assumes the organization's donations/employee associated with the ext_org already exist)

```

PROCEDURE QUERY9

    (Organization_Name_Input IN VARCHAR2, SSN_Input IN NUMBER, Org_Mailing_Address_Input
IN VARCHAR2, Org_Phone_Number_Input IN NUMBER, Org_Contact_Person_Input IN VARCHAR2,
IsOrgAnonymous_Input IN VARCHAR2) AS

    BEGIN

        INSERT INTO EXTERNAL_ORGANIZATION(Organization_Name, SSN,
Org_Mailing_Address, Org_Phone_Number, Org_Contact_Person, IsOrgAnonymous)

            VALUES(Organization_Name_Input, SSN_Input, Org_Mailing_Address_Input,
Org_Phone_Number_Input, Org_Contact_Person_Input, IsOrgAnonymous_Input);

    END;

```

--Query 10: Retrieve the name and phone number of the doctor of a particular client

```

PROCEDURE QUERY10

    (SSN_Input IN NUMBER, Doctors_Name_Output OUT VARCHAR2, Doctors_Number_Output OUT
NUMBER) AS

    BEGIN

        SELECT Doctors_Name INTO Doctors_Name_Output

```

```

FROM Client
WHERE Client.SSN = SSN_Input;

SELECT Doctors_Number INTO Doctors_Number_Output
FROM Client
WHERE Client.SSN = SSN_Input;

END;

```

--Query 11: Retrieve the total amount of expenses charged by each employee for a particular period of time. The list should be sorted by the total amount of expenses

```

PROCEDURE QUERY11
(Expense_Date_Input IN NUMBER, Result_Table_Output OUT SYS_REFCURSOR) AS
BEGIN
    OPEN Result_Table_Output FOR
    SELECT SSN, SUM(Expense_Amount)
    FROM Expense
    WHERE Expense.EXPENSE_DATE = Expense_Date_Input
    GROUP BY SSN
    ORDER BY SUM(Expense_Amount);

END;

```

--Query 12: Retrieve the list of volunteers that are members of teams that support a particular client

```

PROCEDURE QUERY12
(SSN_Input IN NUMBER, Result_Table_Output OUT SYS_REFCURSOR) AS
BEGIN
    OPEN Result_Table_Output FOR
    SELECT SSN, Team_Name
    FROM Serves_on
    WHERE Team_Name IN (SELECT Team_Name FROM Cares_for

```

```

WHERE Serves_on.Team_Name = Cares_for.Team_Name AND SSN IN (SELECT SSN FROM Client
WHERE Cares_for.SSN = Client.SSN AND Client.SSN = SSN_Input));

END;

```

--Query 13: Retrieve the names and contact information of the clients that are supported by teams sponsored by an organization whose name starts with a letter between B and K. The client list should be sorted by name

```

PROCEDURE QUERY13
(Result_Table_Output OUT SYS_REFCURSOR) AS
BEGIN
    OPEN Result_Table_Output FOR
    SELECT *
    FROM CONTACT_INFO
    FULL OUTER JOIN PERSON ON CONTACT_INFO.EMAIL_ADDRESS = PERSON.EMAIL_ADDRESS
    WHERE SSN IN (SELECT SSN FROM CARES_FOR WHERE PERSON.SSN = CARES_FOR.SSN AND
    TEAM_NAME IN (SELECT TEAM_NAME FROM MAY_SPONSOR WHERE
    CARES_FOR.TEAM_NAME = MAY_SPONSOR.TEAM_NAME AND
    ORGANIZATION_NAME >= 'B' AND ORGANIZATION_NAME <= 'K'))
    ORDER BY NAME;
END;

```

--Query 14: 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

```

PROCEDURE QUERY14
(Donation_Result_Table_Output OUT SYS_REFCURSOR, Name_Result_Table_Output OUT
SYS_REFCURSOR, Isanon_Result_Table_Output OUT SYS_REFCURSOR) AS
BEGIN
    -- This cursor finds the sum of donations
    OPEN Donation_Result_Table_Output FOR
    SELECT SUM(DONATION_AMOUNT)

```

```

FROM DONATION
WHERE SSN IN (SELECT SSN FROM DONOR WHERE DONOR.SSN = DONATION.SSN AND
SSN IN (SELECT SSN FROM EMPLOYEE WHERE EMPLOYEE.SSN = DONOR.SSN))
ORDER BY SUM(DONATION_AMOUNT);

```

-- This cursor finds the name

```

OPEN Name_Result_Table_Output FOR
SELECT NAME
FROM PERSON
WHERE SSN IN
(SELECT SSN FROM EMPLOYEE WHERE EMPLOYEE.SSN = PERSON.SSN AND
SSN IN (SELECT SSN FROM DONOR WHERE DONOR.SSN = EMPLOYEE.SSN));

```

-- This cursor finds the isDonorAnon tag

```

OPEN Isanon_Result_Table_Output FOR
SELECT ISDONORANONYMOUS
FROM DONOR
WHERE SSN IN (SELECT SSN FROM EMPLOYEE WHERE EMPLOYEE.SSN = DONOR.SSN);
END;

```

--Query 15: For each team, retrieve the name and associated contact information of the volunteer that has worked the most total hours between March and June

```

PROCEDURE QUERY15
(Result_Table_Output OUT SYS_REFCURSOR) AS
BEGIN
    OPEN Result_Table_Output FOR
SELECT
q0.*, q1.NAME, q4.TEAM_NAME, q4.MAXSUM
FROM

```

```

        (SELECT * FROM CONTACT_INFO) q0
        INNER JOIN
        (SELECT SSN, NAME, EMAIL_ADDRESS FROM PERSON) q1
        ON q0.EMAIL_ADDRESS = q1.EMAIL_ADDRESS
        INNER JOIN
        (SELECT SSN FROM VOLUNTEER) q2
        ON q1.SSN = q2.SSN
        INNER JOIN
        (SELECT SSN, TEAM_NAME,
        MARCH_HOURS_WORKED+APRIL_HOURS_WORKED+MAY_HOURS_WORKED+JUNE_HOURS_WORKED
        AS SUM FROM SERVES_ON) q3
        ON q2.SSN = q3.SSN
        JOIN (SELECT TEAM_NAME,
        MAX(MARCH_HOURS_WORKED+APRIL_HOURS_WORKED+MAY_HOURS_WORKED+JUNE_HOURS_WOR
        KED) AS MAXSUM FROM SERVES_ON GROUP BY TEAM_NAME) q4
        ON q3.SUM = q4.MAXSUM;

    END;

```

--Query 16: Increase the salary by 10% of all employees to whom more than one team must report.

```

    PROCEDURE QUERY16 AS
    BEGIN
        UPDATE EMPLOYEE
        SET SALARY = (CASE
            WHEN (SELECT COUNT(SSN) FROM REPORTS_TO WHERE EMPLOYEE.SSN = REPORTS_TO.SSN) >
            1 THEN SALARY*1.1
            WHEN (SELECT COUNT(SSN) FROM REPORTS_TO WHERE EMPLOYEE.SSN = REPORTS_TO.SSN)
            <= 1 THEN SALARY
        END);
    END;

```

--Query 17: Delete all clients who do not have health insurance and whose value of importance for transportation is less than 5

```

PROCEDURE QUERY17 AS
BEGIN
    DELETE FROM CLIENT
        WHERE (CLIENT.SSN IN (SELECT SSN FROM NEED WHERE CLIENT.SSN = NEED.SSN AND
NEED.NEED = 'Transportation' AND NEED.VALUE_OF_IMPORTANCE < 5))
        AND (CLIENT.SSN NOT IN (SELECT SSN FROM INSURANCE_POLICY WHERE PROVIDER_TYPE =
'Health'));
END;
```

--Query 18 calls Query 1

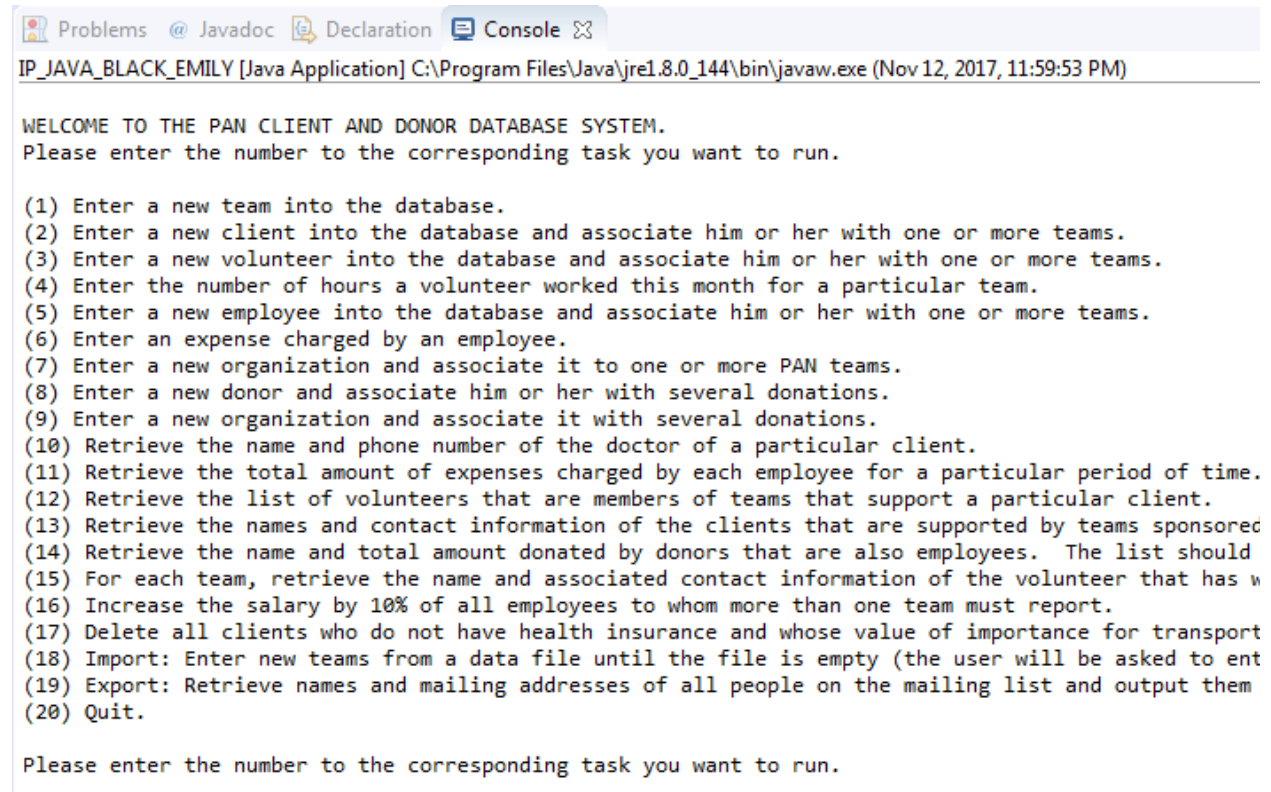
--Query 19: Export: Retrieve names and mailing addresses of all people on the mailing list and output them to a data file instead of screen (the user should be asked to enter the output file name)

```

PROCEDURE QUERY19
(Result_Table_Output OUT SYS_REFCURSOR) AS
BEGIN
    OPEN Result_Table_Output FOR
        SELECT PERSON.NAME, CONTACT_INFO.MAILING_ADDRESS
        FROM PERSON,CONTACT_INFO
        WHERE PERSON.EMAIL_ADDRESS = CONTACT_INFO.EMAIL_ADDRESS AND
PERSON.ISONMAILINGLIST = 'Y';
END;
```

END IP_SQL_BLACK_EMILY;

Java Program



```

Problems @ Javadoc Declaration Console
IP_JAVA_BLACK_EMILY [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Nov 12, 2017, 11:59:53 PM)

WELCOME TO THE PAN CLIENT AND DONOR DATABASE SYSTEM.
Please enter the number to the corresponding task you want to run.

(1) Enter a new team into the database.
(2) Enter a new client into the database and associate him or her with one or more teams.
(3) Enter a new volunteer into the database and associate him or her with one or more teams.
(4) Enter the number of hours a volunteer worked this month for a particular team.
(5) Enter a new employee into the database and associate him or her with one or more teams.
(6) Enter an expense charged by an employee.
(7) Enter a new organization and associate it to one or more PAN teams.
(8) Enter a new donor and associate him or her with several donations.
(9) Enter a new organization and associate it with several donations.
(10) Retrieve the name and phone number of the doctor of a particular client.
(11) Retrieve the total amount of expenses charged by each employee for a particular period of time.
(12) Retrieve the list of volunteers that are members of teams that support a particular client.
(13) Retrieve the names and contact information of the clients that are supported by teams sponsored
(14) Retrieve the name and total amount donated by donors that are also employees. The list should
(15) For each team, retrieve the name and associated contact information of the volunteer that has w
(16) Increase the salary by 10% of all employees to whom more than one team must report.
(17) Delete all clients who do not have health insurance and whose value of importance for transport
(18) Import: Enter new teams from a data file until the file is empty (the user will be asked to ent
(19) Export: Retrieve names and mailing addresses of all people on the mailing list and output them
(20) Quit.

Please enter the number to the corresponding task you want to run.

```

```

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.sql.*;

import java.util.Scanner;

import oracle.jdbc.OracleTypes;

/** Package obtained from http://opencsv.sourceforge.net/ */

import au.com.bytecode.opencsv.CSVReader;

import au.com.bytecode.opencsv.CSVWriter;

/**

* A CLIENT & DONOR DATABASE SYSTEM

```



```

* @author: Emily Black
* @date: 11/13/17
*/
public class IP_JAVA_BLACK_EMILY {
    public static void main(String[] args) throws SQLException {

        // Step 1. Load a database driver:
        try {
            Class.forName("oracle.jdbc.OracleDriver");
        }
        catch(Exception x)
        {
            System.out.println( "Unable to load the driver class!" );
        }

        // Step 2. Create an Oracle JDBC Connection:
        Connection conn = DriverManager.getConnection ( "host", "username", "password");

        // Step 3. Create a new scanner/notDone variable:
        Scanner sc = new Scanner(System.in);
        boolean notDone = true;

        // Step 4. Print the client & donor database system menu:
        String cdMenu =
            "WELCOME TO THE PAN CLIENT AND DONOR DATABASE SYSTEM.\n"
            + "Please enter the number to the corresponding task you want to run.\n"
            + "\n"
            + "(1) Enter a new team into the database.\n"
            + "(2) Enter a new client into the database and associate him or her with one or more teams.\n"

```

```

+ "(3) Enter a new volunteer into the database and associate him or her with one or more teams.\n"
+ "(4) Enter the number of hours a volunteer worked this month for a particular team.\n"
+ "(5) Enter a new employee into the database and associate him or her with one or more
teams.\n"
+ "(6) Enter an expense charged by an employee.\n"
+ "(7) Enter a new organization and associate it to one or more PAN teams.\n"
+ "(8) Enter a new donor and associate him or her with several donations.\n"
+ "(9) Enter a new organization and associate it with several donations.\n"
+ "(10) Retrieve the name and phone number of the doctor of a particular client.\n"
+ "(11) Retrieve the total amount of expenses charged by each employee for a particular period of
time. The list should be sorted by the total amount of expenses\n"
+ "(12) Retrieve the list of volunteers that are members of teams that support a particular client.\n"
+ "(13) Retrieve the names and contact information of the clients that are supported by teams
sponsored by an organization whose name starts with a letter between B and K. The client list should be
sorted by name.\n"
+ "(14) Retrieve the name and total amount donated by 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.\n"
+ "(15) For each team, retrieve the name and associated contact information of the volunteer that
has worked the most total hours between March and June.\n"
+ "(16) Increase the salary by 10% of all employees to whom more than one team must report.\n"
+ "(17) Delete all clients who do not have health insurance and whose value of importance for
transportation is less than 5.\n"
+ "(18) Import: Enter new teams from a data file until the file is empty (the user will be asked to
enter the input file name).\n"
+ "(19) 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 will be asked to enter the output file name).\n"
+ "(20) Quit.";

System.out.println("\n" + cdMenu + "\n");

// Step 5. Begin reading input from user:
do {

```

```

System.out.println("Please enter the number to the corresponding task you want to run.");

int taskToRun = Integer.parseInt(sc.nextLine());

String[] parsedInput;

String userInput;

if (taskToRun == 1) {

    System.out.println("Please enter your input in the following format: Team name, Team type,
Date of formation (Ex: 121495 = Dec. 14th, 1995), Team lead");

    userInput = sc.nextLine();

    parsedInput = userInput.split(" ");

    QUERY1(conn, parsedInput[0], parsedInput[1], parsedInput[2], parsedInput[3]);

}

else if (taskToRun == 2) {

    System.out.println("Please enter your input in the following format: Email address, Mailing
address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB (Ex: 121495 =
Dec. 14th, 1995), Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external
organization (Y or N), Doctors name, Attorneys name, Doctors number, Attorneys number, Date of
assignment, Team name, Is active (Y or N)");

    userInput = sc.nextLine();

    parsedInput = userInput.split(" ");

    QUERY2(conn, parsedInput[0], parsedInput[1], parsedInput[2], parsedInput[3], parsedInput[4],
parsedInput[5], parsedInput[6], parsedInput[7], parsedInput[8], parsedInput[9], parsedInput[10],
parsedInput[11], parsedInput[12], parsedInput[13], parsedInput[14], parsedInput[15], parsedInput[16],
parsedInput[17], parsedInput[18], parsedInput[19]);

}

else if (taskToRun == 3) {

    System.out.println("Please enter your input in the following format: Email address input,
Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB,
Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external org. (Y or N), Join date,
Training date, Training location, Team name, Is active (Y or N), Jan hours worked, Feb hours worked,

```

March hours worked, April hours worked, May hours worked, June hours worked, July hours worked, Aug hours worked, Sept hours worked, Oct hours worked, Nov hours worked, Dec hours worked");

```
    userInput = sc.nextLine();
```

```
    parsedInput = userInput.split(" ");
```

```
    QUERY3(conn, parsedInput[0], parsedInput[1], parsedInput[2], parsedInput[3], parsedInput[4],
    parsedInput[5], parsedInput[6], parsedInput[7], parsedInput[8], parsedInput[9], parsedInput[10],
    parsedInput[11], parsedInput[12], parsedInput[13], parsedInput[14], parsedInput[15], parsedInput[16],
    parsedInput[17], parsedInput[18], parsedInput[19], parsedInput[20], parsedInput[21], parsedInput[22],
    parsedInput[23], parsedInput[24], parsedInput[25], parsedInput[26], parsedInput[27], parsedInput[28],
    parsedInput[29]);
```

```
    }
```

```
    else if (taskToRun == 4) {
```

System.out.println("Please enter your input in the following format: SSN, Jan hours worked, Feb hours worked, March hours worked, April hours worked, May hours worked, June hours worked, July hours worked, Aug hours worked, Sept hours worked, Oct hours worked, Nov hours worked, Dec hours worked");

```
    userInput = sc.nextLine();
```

```
    parsedInput = userInput.split(" ");
```

```
    QUERY4(conn, parsedInput[0], parsedInput[1], parsedInput[2], parsedInput[3], parsedInput[4],
    parsedInput[5], parsedInput[6], parsedInput[7], parsedInput[8], parsedInput[9], parsedInput[10],
    parsedInput[11], parsedInput[12]);
```

```
    }
```

```
    else if (taskToRun == 5) {
```

System.out.println("Please enter your input in the following format: Email address, Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB (Ex: 121495 = Dec. 14th, 1995), Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external organization (Y or N), Salary, Martial status, Hire date, Team name, Date of report (Ex: 121495 = Dec. 14th, 1995), Description");

```
    userInput = sc.nextLine();
```

```
    parsedInput = userInput.split(" ");
```

```
    QUERY5(conn, parsedInput[0], parsedInput[1], parsedInput[2], parsedInput[3], parsedInput[4],
    parsedInput[5], parsedInput[6], parsedInput[7], parsedInput[8], parsedInput[9], parsedInput[10],
```

```
parsedInput[11], parsedInput[12], parsedInput[13], parsedInput[14], parsedInput[15], parsedInput[16],
parsedInput[17], parsedInput[18]);
```

```
} else if (taskToRun == 6) {
```

```
    System.out.println("Please enter your input in the following format: SSN, Expense date (Ex:
121495 = Dec. 14th, 1995), Expense amount, Expense description");
```

```
    userInput = sc.nextLine();
```

```
    parsedInput = userInput.split(" ");
```

```
    QUERY6(conn, parsedInput[0], parsedInput[1], parsedInput[2], parsedInput[3]);
```

```
}
```

```
else if (taskToRun == 7) {
```

```
    System.out.println("Please enter your input in the following format: Organization name, SSN,
Organization mailing address, Organization phone number, Organization contact person, Is org
anonymous (Y or N), Team name");
```

```
    userInput = sc.nextLine();
```

```
    parsedInput = userInput.split(" ");
```

```
    QUERY7(conn, parsedInput[0], parsedInput[1], parsedInput[2], parsedInput[3], parsedInput[4],
parsedInput[5], parsedInput[6]);
```

```
}
```

```
else if (taskToRun == 8) {
```

```
    System.out.println("Please enter your input in the following format: Email address, Mailing
address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB (Ex: 121495 =
Dec. 14th, 1995), Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external
organization (Y or N), Is donor anonymous (Y or N)");
```

```
    userInput = sc.nextLine();
```

```
    parsedInput = userInput.split(" ");
```

```
    QUERY8(conn, parsedInput[0], parsedInput[1], parsedInput[2], parsedInput[3], parsedInput[4],
parsedInput[5], parsedInput[6], parsedInput[7], parsedInput[8], parsedInput[9], parsedInput[10],
parsedInput[11], parsedInput[12], parsedInput[13]);
```

```
}
```

```

else if (taskToRun == 9) {

    System.out.println("Please enter your input in the following format: Organization name, SSN,
Organization mailing address, Organization phone number, Organization contact person, Is org
anonymous (Y or N)");

    userInput = sc.nextLine();

    parsedInput = userInput.split(" ");

    QUERY9(conn, parsedInput[0], parsedInput[1], parsedInput[2], parsedInput[3], parsedInput[4],
parsedInput[5]);

}

```

```

else if (taskToRun == 10) {

    System.out.println("Please enter your input in the following format: SSN");

    userInput = sc.nextLine();

    parsedInput = userInput.split(" ");

    QUERY10(conn, parsedInput[0]);

}

```

```

else if (taskToRun == 11) {

    System.out.println("Please enter your input in the following format: Expense date");

    userInput = sc.nextLine();

    parsedInput = userInput.split(" ");

    QUERY11(conn, parsedInput[0]);

}

```

```

else if (taskToRun == 12) {

    System.out.println("Please enter your input in the following format: SSN");

    userInput = sc.nextLine();

    parsedInput = userInput.split(" ");

    QUERY12(conn, parsedInput[0]);

}

```

```
}
```

```
else if (taskToRun == 13) {
```

```
    // Query takes no input
```

```
    QUERY13(conn);
```

```
}
```

```
else if (taskToRun == 14) {
```

```
    // Query takes no input
```

```
    QUERY14(conn);
```

```
}
```

```
else if (taskToRun == 15) {
```

```
    // Query takes no input
```

```
    QUERY15(conn);
```

```
}
```

```
else if (taskToRun == 16) {
```

```
    // Query takes no input
```

```
    QUERY16(conn);
```

```
}
```

```
else if (taskToRun == 17) {
```

```
    // Query takes no input
```

```
    QUERY17(conn);
```

```
}
```

```
else if (taskToRun == 18) {
```

System.out.println("Please enter the name of your CSV data file to import (Ex: myInputFile.csv). Please make sure your file is located in the project directory.");

userInput = sc.nextLine();

System.out.println(userInput);

QUERY18(conn, userInput);

}

else if (taskToRun == 19) {

System.out.println("Please enter the name of your CSV data file to export (Ex: myExportFile.csv). The file will be placed in the project directory.");

userInput = sc.nextLine();

QUERY19(conn, userInput);

}

else if (taskToRun == 20) {

System.out.println("\n" + "CLOSING THE PAN CLIENT AND DONOR DATABASE SYSTEM, GOODBYE.");

notDone = false;

}

else {

System.out.print("\n");

System.out.println(taskToRun + " isn't a valid task number. Try again!");

System.out.print("\n");

}

} while (notDone);

sc.close();

conn.close();

}

// Step 6. Call specified SQL procedure:

```

/** This method calls the SQL procedure QUERY1
 * @task: Enter a new team into the database.
 * @params: Team name, Team type, Date of formation, Team lead
 */
public static void QUERY1(Connection conn, String arg1, String arg2, String arg3, String arg4) {
    try {
        CallableStatement stmt = conn.prepareCall("{call QUERY1(?, ?, ?, ?)}");
        stmt.setString(1, arg1);
        stmt.setString(2, arg2);
        stmt.setString(3, arg3);
        stmt.setString(4, arg4);
        stmt.execute();
        System.out.println("New team successfully created!");
    } catch (SQLException e) {
        System.out.println("(ERROR) SQL exception: " + e.getMessage());
    }
}

```

```

/** This method calls the SQL procedure QUERY2
 * @task: Enter a new client into the database and associate him or her with one or more teams.
 * @params: Email address, Mailing address, Home phone number, Work phone number, Cell phone
number, SSN, Name, DoB (Ex: 121495 = Dec. 14th, 1995), Race, Gender,
 * Profession, Is on mailing list (Y or N), Is affiliated with external organization (Y or N), Doctors name,
Attorneys name, Doctors number, Attorneys number,
 * Date of assignment, Team name, Is active (Y or N)
 */

```

```

public static void QUERY2(Connection conn, String arg1, String arg2, String arg3, String arg4, String
arg5, String arg6, String arg7, String arg8, String arg9, String arg10, String arg11, String arg12, String
arg13, String arg14, String arg15, String arg16, String arg17, String arg18, String arg19, String arg20) {
    try {
        CallableStatement stmt = conn.prepareCall("{call QUERY2(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?, ?, ?)}");
        stmt.setString(1, arg1);
        stmt.setString(2, arg2);
        stmt.setString(3, arg3);
        stmt.setString(4, arg4);
        stmt.setString(5, arg5);
        stmt.setString(6, arg6);
        stmt.setString(7, arg7);
        stmt.setString(8, arg8);
        stmt.setString(9, arg9);
        stmt.setString(10, arg10);
        stmt.setString(11, arg11);
        stmt.setString(12, arg12);
        stmt.setString(13, arg13);
        stmt.setString(14, arg14);
        stmt.setString(15, arg15);
        stmt.setString(16, arg16);
        stmt.setString(17, arg17);
        stmt.setString(18, arg18);
        stmt.setString(19, arg19);
        stmt.setString(20, arg20);
        stmt.execute();
        System.out.println("New client successfully created!");
    } catch (SQLException e) {
        System.out.println("(ERROR) SQL exception: " + e.getMessage());
    }
}

```

```

    }
}

/**This method calls the SQL procedure QUERY3

* @task: Enter a new volunteer into the database and associate him or her with one or more teams.

* @params: Email address input, Mailing address, Home phone number, Work phone number, Cell
phone number, SSN, Name, DoB, Race, Gender, Profession,

* Is on mailing list (Y or N), Is affiliated with external org. (Y or N), Join date, Training date, Training
location, Team name, Is active (Y or N),

* Jan hours worked, Feb hours worked, March hours worked, April hours worked, May hours worked,
June hours worked, July hours worked, Aug hours worked,

* Sept hours worked, Oct hours worked, Nov hours worked, Dec hours worked)

*/

public static void QUERY3(Connection conn, String arg1, String arg2, String arg3, String arg4, String
arg5, String arg6, String arg7, String arg8, String arg9, String arg10, String arg11, String arg12, String
arg13, String arg14, String arg15, String arg16, String arg17, String arg18, String arg19, String arg20,
String arg21, String arg22, String arg23, String arg24, String arg25, String arg26, String arg27, String
arg28, String arg29, String arg30) {

    try {

        CallableStatement stmt = conn.prepareCall("{call QUERY3(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)}");

        stmt.setString(1, arg1);

        stmt.setString(2, arg2);

        stmt.setString(3, arg3);

        stmt.setString(4, arg4);

        stmt.setString(5, arg5);

        stmt.setString(6, arg6);

        stmt.setString(7, arg7);

        stmt.setString(8, arg8);

        stmt.setString(9, arg9);

        stmt.setString(10, arg10);

        stmt.setString(11, arg11);

```

```

        stmt.setString(12, arg12);
        stmt.setString(13, arg13);
        stmt.setString(14, arg14);
        stmt.setString(15, arg15);
        stmt.setString(16, arg16);
        stmt.setString(17, arg17);
        stmt.setString(18, arg18);
        stmt.setString(19, arg19);
        stmt.setString(20, arg20);
        stmt.setString(21, arg21);
        stmt.setString(22, arg22);
        stmt.setString(23, arg23);
        stmt.setString(24, arg24);
        stmt.setString(25, arg25);
        stmt.setString(26, arg26);
        stmt.setString(27, arg27);
        stmt.setString(28, arg28);
        stmt.setString(29, arg29);
        stmt.setString(30, arg30);

        stmt.execute();

        System.out.println("New client successfully created!");
    } catch (SQLException e) {
        System.out.println("(ERROR) SQL exception: " + e.getMessage());
    }
}

/**
 * This method calls the SQL procedure QUERY4
 * @task: Enter the number of hours a volunteer worked this month for a particular team.

```

* @params: SSN, Jan hours worked, Feb hours worked, March hours worked, April hours worked, May hours worked, June hours worked,

* July hours worked, Aug hours worked, Sept hours worked, Oct hours worked, Nov hours worked, Dec hours worked)

*/

```
public static void QUERY4(Connection conn, String arg1, String arg2, String arg3, String arg4, String
arg5, String arg6, String arg7, String arg8, String arg9, String arg10, String arg11, String arg12, String
arg13) {
```

```
    try {
```

```
        CallableStatement stmt = conn.prepareCall("{call QUERY4(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)}");
```

```
        stmt.setString(1, arg1);
```

```
        stmt.setString(2, arg2);
```

```
        stmt.setString(3, arg3);
```

```
        stmt.setString(4, arg4);
```

```
        stmt.setString(5, arg5);
```

```
        stmt.setString(6, arg6);
```

```
        stmt.setString(7, arg7);
```

```
        stmt.setString(8, arg8);
```

```
        stmt.setString(9, arg9);
```

```
        stmt.setString(10, arg10);
```

```
        stmt.setString(11, arg11);
```

```
        stmt.setString(12, arg12);
```

```
        stmt.setString(13, arg13);
```

```
        stmt.execute();
```

```
        System.out.println("New hours a volunteer worked successfully created!");
```

```
    } catch (SQLException e) {
```

```
        System.out.println("(ERROR) SQL exception: " + e.getMessage());
```

```
    }
```

```
}
```

```

/**
 * This method calls the SQL procedure QUERY5
 *
 * @task: Enter a new employee into the database and associate him or her with one or more teams.
 *
 * @params: Email address, Mailing address, Home phone number, Work phone number, Cell phone
number, SSN, Name, DoB (Ex: 121495 = Dec. 14th, 1995), Race, Gender,
 *
 * Profession, Is on mailing list (Y or N), Is affiliated with external organization (Y or N), Salary, Martial
status, Hire date, Team name,
 *
 * Date of report (Ex: 121495 = Dec. 14th, 1995), Description
 */

public static void QUERY5(Connection conn, String arg1, String arg2, String arg3, String arg4, String
arg5, String arg6, String arg7, String arg8, String arg9, String arg10, String arg11, String arg12, String
arg13, String arg14, String arg15, String arg16, String arg17, String arg18, String arg19) {

    try {

        CallableStatement stmt = conn.prepareCall("{call QUERY5(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?, ?)}");

        stmt.setString(1, arg1);

        stmt.setString(2, arg2);

        stmt.setString(3, arg3);

        stmt.setString(4, arg4);

        stmt.setString(5, arg5);

        stmt.setString(6, arg6);

        stmt.setString(7, arg7);

        stmt.setString(8, arg8);

        stmt.setString(9, arg9);

        stmt.setString(10, arg10);

        stmt.setString(11, arg11);

        stmt.setString(12, arg12);

        stmt.setString(13, arg13);

        stmt.setString(14, arg14);

        stmt.setString(15, arg15);
    }
}

```

```

        stmt.setString(16, arg16);
        stmt.setString(17, arg17);
        stmt.setString(18, arg18);
        stmt.setString(19, arg19);
        stmt.execute();
        System.out.println("New employee successfully created!");
    } catch (SQLException e) {
        System.out.println("(ERROR) SQL exception: " + e.getMessage());
    }
}

/**
 * This method calls the SQL procedure QUERY6
 * @task: Enter an expense charged by an employee.
 * @params: SSN, Expense date (Ex: 121495 = Dec. 14th, 1995), Expense amount, Expense description
 */
public static void QUERY6(Connection conn, String arg1, String arg2, String arg3, String arg4) {
    try {
        CallableStatement stmt = conn.prepareCall("{call QUERY6(?, ?, ?, ?)}");
        stmt.setString(1, arg1);
        stmt.setString(2, arg2);
        stmt.setString(3, arg3);
        stmt.setString(4, arg4);
        stmt.execute();
        System.out.println("New expense successfully created!");
    } catch (SQLException e) {
        System.out.println("(ERROR) SQL exception: " + e.getMessage());
    }
}

```

```

/**
 * This method calls the SQL procedure QUERY7
 *
 * @task: Enter a new organization and associate it to one or more PAN teams.
 *
 * @params: Organization name, SSN, Organization mailing address, Organization phone number,
Organization contact person, Is org anonymous (Y or N), Team name
 */

```

```

public static void QUERY7(Connection conn, String arg1, String arg2, String arg3, String arg4, String
arg5, String arg6, String arg7) {

```

```

    try {

```

```

        CallableStatement stmt = conn.prepareCall("{call QUERY7(?, ?, ?, ?, ?, ?, ?)}");

```

```

        stmt.setString(1, arg1);

```

```

        stmt.setString(2, arg2);

```

```

        stmt.setString(3, arg3);

```

```

        stmt.setString(4, arg4);

```

```

        stmt.setString(5, arg5);

```

```

        stmt.setString(6, arg6);

```

```

        stmt.setString(7, arg7);

```

```

        stmt.execute();

```

```

        System.out.println("New organization successfully created!");

```

```

    } catch (SQLException e) {

```

```

        System.out.println("(ERROR) SQL exception: " + e.getMessage());

```

```

    }

```

```

}

```

```

/**

```

```

 * This method calls the SQL procedure QUERY8

```

```

 * @task: Enter a new donor and associate him or her with several donations.

```

```

 * @params: Email address, Mailing address, Home phone number, Work phone number, Cell phone
number, SSN, Name, DoB (Ex: 121495 = Dec. 14th, 1995), Race, Gender,

```


* Profession, Is on mailing list (Y or N), Is affiliated with external organization (Y or N), Is donor anonymous (Y or N)

*/

```
public static void QUERY8(Connection conn, String arg1, String arg2, String arg3, String arg4, String
arg5, String arg6, String arg7, String arg8, String arg9, String arg10, String arg11, String arg12, String
arg13, String arg14) {
```

```
    try {
```

```
        CallableStatement stmt = conn.prepareCall("{call QUERY8(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)}");
```

```
        stmt.setString(1, arg1);
```

```
        stmt.setString(2, arg2);
```

```
        stmt.setString(3, arg3);
```

```
        stmt.setString(4, arg4);
```

```
        stmt.setString(5, arg5);
```

```
        stmt.setString(6, arg6);
```

```
        stmt.setString(7, arg7);
```

```
        stmt.setString(8, arg8);
```

```
        stmt.setString(9, arg9);
```

```
        stmt.setString(10, arg10);
```

```
        stmt.setString(11, arg11);
```

```
        stmt.setString(12, arg12);
```

```
        stmt.setString(13, arg13);
```

```
        stmt.setString(14, arg14);
```

```
        stmt.execute();
```

```
        System.out.println("New donor successfully created!");
```

```
    } catch (SQLException e) {
```

```
        System.out.println("(ERROR) SQL exception: " + e.getMessage());
```

```
    }
```

```
}
```

```
/**
```

* This method calls the SQL procedure QUERY9

* @task: Enter a new organization and associate it with several donations.

* @params: Organization name, SSN, Organization mailing address, Organization phone number, Organization contact person, Is org anonymous (Y or N)

*/

```
public static void QUERY9(Connection conn, String arg1, String arg2, String arg3, String arg4, String arg5, String arg6) {
```

```
    try {
```

```
        CallableStatement stmt = conn.prepareCall("{call QUERY9(?, ?, ?, ?, ?, ?)}");
```

```
        stmt.setString(1, arg1);
```

```
        stmt.setString(2, arg2);
```

```
        stmt.setString(3, arg3);
```

```
        stmt.setString(4, arg4);
```

```
        stmt.setString(5, arg5);
```

```
        stmt.setString(6, arg6);
```

```
        stmt.execute();
```

```
        System.out.println("New organization successfully created!");
```

```
    } catch (SQLException e) {
```

```
        System.out.println("(ERROR) SQL exception: " + e.getMessage());
```

```
    }
```

```
}
```

```
/**
```

* This method calls the SQL procedure QUERY10

* @task: Retrieve the name and phone number of the doctor of a particular client.

* @params: SSN

*/

```
public static void QUERY10(Connection conn, String arg1) {
```

```
    try {
```

```

CallableStatement stmt = conn.prepareCall("{call QUERY10(?, ?, ?)}");
stmt.setString(1, arg1);
stmt.registerOutParameter(2, Types.VARCHAR);
stmt.registerOutParameter(3, Types.NUMERIC);
stmt.execute();
System.out.println("\n" + "RESULTS:" + "\n");
System.out.println("Name of the client's doctor: " + stmt.getString(2));
System.out.println("Number of the client's doctor: " + stmt.getLong(3) + "\n");
} catch (SQLException e) {
    System.out.println("(ERROR) SQL exception: " + e.getMessage());
}
}

```

```
/**
```

```
 * This method calls the SQL procedure QUERY11
```

```
 * @task: 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.
```

```
 * @params: Expense date
```

```
 */
```

```
public static void QUERY11(Connection conn, String arg1) {
```

```
    try {
```

```
        CallableStatement stmt = conn.prepareCall("{call QUERY11(?, ?)}");
```

```
        stmt.setString(1, arg1);
```

```
        stmt.registerOutParameter(2, OracleTypes.CURSOR);
```

```
        stmt.execute();
```

```
        System.out.println("\n" + "RESULTS:" + "\n");
```

```
        System.out.println("EXPENSE AMOUNT, SSN");
```

```
        ResultSet rs = (ResultSet) stmt.getObject(2);
```

```

        while(rs.next()) {
            System.out.println("$" + rs.getString("SUM(EXPENSE_AMOUNT)") + ", " + rs.getString("SSN"));
        }

        System.out.print("\n");
    } catch (SQLException e) {
        System.out.println("(ERROR) SQL exception: " + e.getMessage());
    }
}

```

```

/**
 * This method calls the SQL procedure QUERY12
 * @task: Retrieve the list of volunteers that are members of teams that support a particular client.
 * @params: SSN
 */

```

```

public static void QUERY12(Connection conn, String arg1) {
    try {
        CallableStatement stmt = conn.prepareCall("{call QUERY12(?, ?)}");

        stmt.setString(1, arg1);

        stmt.registerOutParameter(2, OracleTypes.CURSOR);

        stmt.execute();

        System.out.println("\n" + "RESULTS:" + "\n");

        System.out.println("SSN, TEAM NAME");

        ResultSet rs = (ResultSet) stmt.getObject(2);

        while(rs.next()) {
            System.out.println(rs.getString("SSN") + ", " + rs.getString("TEAM_NAME"));
        }

        System.out.print("\n");
    } catch (SQLException e) {
        System.out.println("(ERROR) SQL exception: " + e.getMessage());
    }
}

```

```

    }
}

/**
 * This method calls the SQL procedure QUERY13
 *
 * @task: Retrieve the names and contact information of the clients that are supported by teams
sponsored by an organization whose name starts with a
 *
 * letter between B and K. The client list should be sorted by name.
 *
 * @params: none, user input isn't needed
 */
public static void QUERY13(Connection conn) {
    try {
        CallableStatement stmt = conn.prepareCall("{call QUERY13(?)}");
        stmt.registerOutParameter(1, OracleTypes.CURSOR);
        stmt.execute();
        System.out.println("\n" + "RESULTS:" + "\n");
        System.out.println("NAME, EMAIL ADDRESS, MAILING ADDRESS, HOME PHONE, WORK PHONE,
CELL PHONE");
        ResultSet rs = (ResultSet) stmt.getObject(1);
        while(rs.next()) {
            System.out.println(rs.getString("NAME") + ", " + rs.getString("EMAIL_ADDRESS") + ", " +
rs.getString("MAILING_ADDRESS") + ", " + rs.getString("HOME_PHONE_NUMBER") + ", " +
rs.getString("WORK_PHONE_NUMBER") + ", " + rs.getString("CELL_PHONE_NUMBER"));
        }
        System.out.print("\n");
    } catch (SQLException e) {
        System.out.println("(ERROR) SQL exception: " + e.getMessage());
    }
}

```

```

/**
 * This method calls the SQL procedure QUERY14
 *
 * @task: 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.
 *
 * @params: none, user input isn't needed
 */
public static void QUERY14(Connection conn) {
    try {
        CallableStatement stmt = conn.prepareCall("{call QUERY14(?, ?, ?)}");
        stmt.registerOutParameter(1, OracleTypes.CURSOR);
        stmt.registerOutParameter(2, OracleTypes.CURSOR);
        stmt.registerOutParameter(3, OracleTypes.CURSOR);
        stmt.execute();

        System.out.println("\n" + "RESULTS:" + "\n");
        ResultSet rs1 = (ResultSet) stmt.getObject(1);
        while(rs1.next()) {
            System.out.println("Total Amount of Donations: $" +
rs1.getString("SUM(DONATION_AMOUNT)"));
        }

        ResultSet rs2 = (ResultSet) stmt.getObject(2);
        System.out.print("Contributors: ");
        while(rs2.next()) {
            System.out.print(rs2.getString("NAME") + ", ");
        }

        ResultSet rs3 = (ResultSet) stmt.getObject(3);
        System.out.print("\n");
        System.out.print("Is Donor Anonymous (True or False): ");
        while(rs3.next()) {

```

```

        System.out.print(rs3.getString("ISDONORANONYMOUS") + ", ");
    }

    System.out.println("\n");
} catch (SQLException e) {
    System.out.println("(ERROR) SQL exception: " + e.getMessage());
}
}

/**
 * This method calls the SQL procedure QUERY15
 *
 * @task: For each team, retrieve the name and associated contact information of the volunteer that
has worked the most total hours between March and June.
 *
 * @params: none, user input isn't needed
 */
public static void QUERY15(Connection conn) {
    try {
        CallableStatement stmt = conn.prepareCall("{call QUERY15(?)}");
        stmt.registerOutParameter(1, OracleTypes.CURSOR);
        stmt.execute();

        System.out.println("\n" + "RESULTS:" + "\n");

        System.out.println("EMAIL ADDRESS, MAILING ADDRESS, HOME PHONE, WORK PHONE, CELL
PHONE, NAME, TEAM, HOURS");

        ResultSet rs = (ResultSet) stmt.getObject(1);

        while(rs.next()) {
            System.out.println(rs.getString("EMAIL_ADDRESS") + ", " + rs.getString("MAILING_ADDRESS") +
", " + rs.getString("HOME_PHONE_NUMBER") + ", " + rs.getString("WORK_PHONE_NUMBER") + ", " +
rs.getString("CELL_PHONE_NUMBER") + ", " + rs.getString("NAME") + ", " +
rs.getString("TEAM_NAME") + ", " + rs.getString("MAXSUM"));
        }

        System.out.print("\n");
    }
}

```

```

    } catch (SQLException e) {
        System.out.println("(ERROR) SQL exception: " + e.getMessage());
    }
}

```

```

/**
 * This method calls the SQL procedure QUERY16
 * @task: Increase the salary by 10% of all employees to whom more than one team must report.
 * @params: none, user input isn't needed
 */

```

```

public static void QUERY16(Connection conn) {
    try {
        CallableStatement stmt = conn.prepareCall("{call QUERY16()}");
        stmt.execute();
        System.out.println("\n" + "Salaries successfully updated!" + "\n");
    } catch (SQLException e) {
        System.out.println("(ERROR) SQL exception: " + e.getMessage());
    }
}

```

```

/**
 * This method calls the SQL procedure QUERY17
 * @task: Delete all clients who do not have health insurance and whose value of importance for
transportation is less than 5.

```

```

 * @params: none, user input isn't needed

```

```

 */

```

```

public static void QUERY17(Connection conn) {
    try {
        CallableStatement stmt = conn.prepareCall("{call QUERY17()}");

```



```

        stmt.execute();

        System.out.println("\n" + "Clients successfully updated!" + "\n");
    } catch (SQLException e) {
        System.out.println("(ERROR) SQL exception: " + e.getMessage());
    }
}

/**
 * This method calls the SQL procedure QUERY18
 * @task: Import: Enter new teams from a data file until the file is empty
 * (the user should be asked to enter the input file name).
 * @params: Name of file
 */
public static void QUERY18(Connection conn, String nameOfFile) {
    try {
        // Can call query1, because it already handles adding data into the Team table
        CallableStatement stmt = conn.prepareCall("{call QUERY1(?, ?, ?, ?)}");

        CSVReader reader = null;

        try {
            reader = new CSVReader(new FileReader(nameOfFile));
        } catch (FileNotFoundException e1) {
            e1.printStackTrace();
        }

        String [] nextLine;
        int lineNum = 0;

        // Loop through the entire file, add each record (set) to the batch
        try {
            while ((nextLine = reader.readNext()) != null) {
                lineNum++;
            }
        }
    }
}

```

```

        stmt.setString(1, nextLine[0]);
        stmt.setString(2, nextLine[1]);
        stmt.setDouble(3, Integer.parseInt(nextLine[2]));
        stmt.setString(4, nextLine[3]);

        // Add the record to batch
        stmt.addBatch();
    }
} catch (NumberFormatException e1) {
    e1.printStackTrace();
} catch (IOException e1) {
    e1.printStackTrace();
}

// Start bulk batch insert
int[] totalRecords = new int[lineNum];

try {
    totalRecords = stmt.executeBatch();
} catch (BatchUpdateException e) {
    totalRecords = e.getUpdateCounts();
    System.out.println("(ERROR) INSERTION exception: " + e.getMessage());
}

System.out.println("Total records inserted in bulk from CSV file: " + totalRecords.length);

if (totalRecords.length > 1) {
    System.out.println("New teams successfully imported!" + "\n");
}

else {
    System.out.println("New team successfully imported!" + "\n");
}

try {
    reader.close();

```

```

        } catch (IOException e) {
            e.printStackTrace();
        }
    } catch (SQLException e) {
        System.out.println("(ERROR) IMPORT exception: " + e.getMessage());
    }
}

/**
 * This method calls the SQL procedure QUERY19
 *
 * @task: 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 should be asked to enter the output file name).
 *
 * @params: Name of file
 */
public static void QUERY19(Connection conn, String nameOfFile) {
    try {
        CallableStatement stmt = conn.prepareCall("{call QUERY19(?)}");
        CSVWriter writer = null;

        try {
            writer = new CSVWriter(new FileWriter(nameOfFile));
        } catch (IOException e) {
            e.printStackTrace();
        }

        stmt.registerOutParameter(1, OracleTypes.CURSOR);
        stmt.execute();
        ResultSet rs = (ResultSet) stmt.getObject(1);
        // Writes all data from ResultSet to output file. Includes column names.
        try {

```

```
        writer.writeAll(rs, true);
    } catch (IOException e) {
        e.printStackTrace();
    }

    try {

        writer.close();
    } catch (IOException e) {
        e.printStackTrace();
    }

    System.out.println("\n" + "Data successfully exported to " + nameOfFile + "\n");
} catch (SQLException e) {
    System.out.println("(ERROR) EXPORT exception: " + e.getMessage());
}
}
}
```

Task 6. Java program Execution

6.1. Script file showing the testing of query 1

IP_JAVA_BLACK_EMILY [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Nov 13, 2017, 5:31:33 PM)
 (19) Export: Retrieve names and mailing addresses of all people on the mailing list and
 (20) Quit.

Please enter the number to the corresponding task you want to run.

1

Please enter your input in the following format: Team name, Team type, Date of formation
 Team Yellow, Marketing, 031307, Barbara Jenkins

New team successfully created!

Please enter the number to the corresponding task you want to run.

1

Please enter your input in the following format: Team name, Team type, Date of formation
 Team White, Finance, 050214, Barry Jenkins

New team successfully created!

Please enter the number to the corresponding task you want to run.

1

Please enter your input in the following format: Team name, Team type, Date of formation
 Team Black, Operations, 072709, Jerry Brown

New team successfully created!

Please enter the number to the corresponding task you want to run.

Please enter the number to the corresponding task you want to run.

1

Please enter your input in the following format: Team name, Team type, Date of formation (Ex: 121495 = Dec. 14th, 1995), Team lead

Team Yellow, Marketing, 031307, Barbara Jenkins

New team successfully created!

Please enter the number to the corresponding task you want to run.

1

Please enter your input in the following format: Team name, Team type, Date of formation (Ex: 121495 = Dec. 14th, 1995), Team lead

Team White, Finance, 050214, Barry Jenkins

New team successfully created!

Please enter the number to the corresponding task you want to run.

1

Please enter your input in the following format: Team name, Team type, Date of formation (Ex: 121495 = Dec. 14th, 1995), Team lead

Team Black, Operations, 072709, Jerry Brown

New team successfully created!

Please enter the number to the corresponding task you want to run.

1

Please enter your input in the following format: Team name, Team type, Date of formation (Ex: 121495 = Dec. 14th, 1995), Team lead

Team Brown, Networking, 080114, Donald Jones

New team successfully created! Please enter the number to the corresponding task you want to run.

1

Please enter your input in the following format: Team name, Team type, Date of formation (Ex: 121495 = Dec. 14th, 1995), Team lead

Team Gold, Resources, 072413, Bob Jones

New team successfully created!

| | TEAM_NAME | TEAM_TYPE | DATE_OF_FORMATION | TEAM_LEAD |
|---|-------------|------------|-------------------|-----------------|
| 1 | Team Black | Operations | 72709 | Jerry Brown |
| 2 | Team Brown | Networking | 80114 | Donald Jones |
| 3 | Team Yellow | Marketing | 31307 | Barbara Jenkins |
| 4 | Team White | Finance | 50214 | Barry Jenkins |
| 5 | Team Gold | Resources | 72413 | Bob Jones |

6.2. Script file showing the testing of query 2

```

IP_JAVA_BLACK_EMILY [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Nov 13, 2017, 5:49:57 PM)
Please enter your input in the following format: Email address, Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB (Ex: 121495 = Dec. 14th, 1995), Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external organization (Y or N), Doctors name, Attorneys name, Doctors number, Attorneys number, Date of assignment, Team name, Is active (Y or N)
zack@ou.edu, 8431 West Drive, 5823258522, 0958424855, 2031139482, 744210, Zack Morenio, 0918, White, M, CS, N, N, Dr. Parker, Jack Johnson, 0492395777, 0492831233, 090714, Team Gold, Y
New client successfully created!
Please enter the number to the corresponding task you want to run.
2
Please enter your input in the following format: Email address, Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB (Ex: 121495 = Dec. 14th, 1995), Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external organization (Y or N), Doctors name, Attorneys name, Doctors number, Attorneys number, Date of assignment, Team name, Is active (Y or N)
bill@gmail.edu, Three Partners Place, 0459011922, 0492310422, 0942314232, 567890, Bill Gardner, 0918, White, M, CS, N, N, Dr. Parker, Jack Johnson, 0492395777, 0492831233, 090714, Team Gold, Y
New client successfully created!
Please enter the number to the corresponding task you want to run.
2
Please enter your input in the following format: Email address, Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB (Ex: 121495 = Dec. 14th, 1995), Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external organization (Y or N), Doctors name, Attorneys name, Doctors number, Attorneys number, Date of assignment, Team name, Is active (Y or N)
jerry.m.jones@hotmail.net, Five Partners Place, 0492319322, 0942312332, 0492310442, 987654, Jerry M. Jones, 0918, White, M, CS, N, N, Dr. Parker, Jack Johnson, 0492395777, 0492831233, 090714, Team Gold, Y
New client successfully created!
Please enter the number to the corresponding task you want to run.
2
Please enter your input in the following format: Email address, Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB (Ex: 121495 = Dec. 14th, 1995), Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external organization (Y or N), Doctors name, Attorneys name, Doctors number, Attorneys number, Date of assignment, Team name, Is active (Y or N)
sarah-johnson571@msu.edu, 492 Palace Place, 0492319332, 5828329522, 5842392333, 321098, Sarah Johnson, 0918, White, F, CS, N, N, Dr. Parker, Jack Johnson, 0492395777, 0492831233, 090714, Team Gold, Y
New client successfully created!
Please enter the number to the corresponding task you want to run.

```

Please enter the number to the corresponding task you want to run.

2

Please enter your input in the following format: Email address, Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB (Ex: 121495 = Dec. 14th, 1995), Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external organization (Y or N), Doctors name, Attorneys name, Doctors number, Attorneys number, Date of assignment, Team name, Is active (Y or N)
 emily@ou.edu, 842 Bobs Lane, 0523921423, 8574839284, 0293577382, 123456, Emily Black, 121495, White, F, CS, N, N, Dr. Parker, Jack Johnson, 0492395777, 0492831233, 090714, Team Gold, Y

New client successfully created!

Please enter the number to the corresponding task you want to run.

2

Please enter your input in the following format: Email address, Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB (Ex: 121495 = Dec. 14th, 1995), Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated

with external organization (Y or N), Doctors name, Attorneys name, Doctors number, Attorneys number, Date of assignment, Team name, Is active (Y or N)
 zack@ou.edu, 8431 West Drive, 5823258522, 0958424855, 2031139482, 744210, Zack Morenio, 091893, White, M, EE, Y, N, Dr. Jones, Sadie Brown, 2432432117, 3918418499, 091501, Team Brown, N

New client successfully created!

Please enter the number to the corresponding task you want to run.

2

Please enter your input in the following format: Email address, Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB (Ex: 121495 = Dec. 14th, 1995), Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external organization (Y or N), Doctors name, Attorneys name, Doctors number, Attorneys number, Date of assignment, Team name, Is active (Y or N)

bill@gmail.edu, Three Partners Place, 0459011922, 0492310422, 0942314232, 567890, Bill Gardener, 010476, African American, M, Physics, Y, Y, Dr. Jones, Sadie Brown, 2432432117, 3918418499, 091501, Team Yellow, N

New client successfully created!

Please enter the number to the corresponding task you want to run.

2

Please enter your input in the following format: Email address, Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB (Ex: 121495 = Dec. 14th, 1995), Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external organization (Y or N), Doctors name, Attorneys name, Doctors number, Attorneys number, Date of assignment, Team name, Is active (Y or N)

jerry.m.jones@hotmail.net, Five Partners Place, 0492319322, 0942312332, 0492310442, 987654, Jerry Michael Jones, 062101, Asian, M, Marketing, N, Y, Dr. Jones, Sadie Brown, 2432432117, 3918418499, 091501, Team White, Y

New client successfully created!

Please enter the number to the corresponding task you want to run.

2

Please enter your input in the following format: Email address, Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB (Ex: 121495 = Dec. 14th, 1995), Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external organization (Y or N), Doctors name, Attorneys name, Doctors number, Attorneys number, Date of assignment, Team name, Is active (Y or N)

sarah-johnson571@msu.edu, 492 Palace Place, 0492319332, 5828329522, 5842392333, 321098, Sarah Johnson, 062101, Mexican, F, Human Resources, N, N, Dr. Jones, Sadie Brown, 2432432117, 3918418499, 091501, Team Black, Y

New client successfully created!

| CS4513 x CONTACT_INFO x QUERY2 x | | | | | |
|---|---------------------------|----------------------|-------------------|-------------------|-------------------|
| Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL | | | | | |
| Sort.. Filter: | | | | | |
| | EMAIL_ADDRESS | MAILING_ADDRESS | HOME_PHONE_NUMBER | WORK_PHONE_NUMBER | CELL_PHONE_NUMBER |
| 1 | emily@ou.edu | 842 Bobs Lane | 523921423 | 8574839284 | 293577382 |
| 2 | zack@ou.edu | 8431 West Drive | 5823258522 | 958424855 | 2031139482 |
| 3 | bill@gmail.edu | Three Partners Place | 459011922 | 492310422 | 942314232 |
| 4 | jerry.m.jones@hotmail.net | Five Partners Place | 492319322 | 942312332 | 492310442 |
| 5 | sarah-johnson571@msu.edu | 492 Palace Place | 492319332 | 5828329522 | 5842392333 |
| 6 | Sarah@ou.edu | 123 Branch Dr. | 4053825923 | 9542359432 | 2753242352 |
| 7 | John@ou.edu | 123 Branch Dr. | 4053825923 | 9542359432 | 2753242352 |
| 8 | Bill@ou.edu | 123 Branch Dr. | 4053825923 | 9542359432 | 2753242352 |

CS4513 PERSON QUERY2

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL

Sort.. Filter:

| | SSN | EMAIL_ADDRESS | NAME | DOB | RACE | GENDER | PROFESSION | ISONMAILINGLIST | ISAFF |
|---|--------|----------------|---------------|-------|--------|--------|-------------|-----------------|-------|
| 1 | 123456 | emily@ou.edu | Emily Black | 12... | White | F | CS | N | N |
| 2 | 744210 | zack@ou.edu | Zack Morenio | 91893 | White | M | EE | Y | N |
| 3 | 567890 | bill@gmail.edu | Bill Gardener | 10476 | Afr... | M | Physics | Y | Y |
| 4 | 987654 | jerry.m.jon... | Jerry Mich... | 62101 | Asian | M | Marketing | N | Y |
| 5 | 321098 | sarah-johns... | Sarah Johnson | 62101 | Mex... | F | Human Re... | N | N |

CS4513 CLIENT QUERY2

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL

Sort.. Filter:

| | SSN | DOCTORS_NAME | ATTORNEYS_NAME | DOCTORS_NUMBER | ATTORNEYS_NUMBER | DATE_OF_ASSIGNMENT |
|---|--------|--------------|----------------|----------------|------------------|--------------------|
| 1 | 123456 | Dr. Parker | Jack Johnson | 492395777 | 492831233 | 90714 |
| 2 | 744210 | Dr. Jones | Sadie Brown | 2432432117 | 3918418499 | 91501 |
| 3 | 567890 | Dr. Jones | Sadie Brown | 2432432117 | 3918418499 | 91501 |
| 4 | 987654 | Dr. Jones | Sadie Brown | 2432432117 | 3918418499 | 91501 |
| 5 | 321098 | Dr. Jones | Sadie Brown | 2432432117 | 3918418499 | 91501 |

CS4513 CARES_FOR QUERY2

Columns Data Model Constraints Grants Statistics Triggers Flashback

Sort.. Filter:

| | SSN | TEAM_NAME | ISACTIVE |
|---|--------|-------------|----------|
| 1 | 123456 | Team Gold | Y |
| 2 | 744210 | Team Brown | N |
| 3 | 567890 | Team Yellow | N |
| 4 | 987654 | Team White | Y |
| 5 | 321098 | Team Black | Y |

6.3. Script file showing the testing of query 3

```

IP_JAVA_BLACK_EMILY [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Nov 13, 2017, 6:27:20 PM)
Please enter your input in the following format: Email address input, Mailing address, Home phone number, Name
jack@gmail.com, 764 Crest Dr., 0593021132, 0492310492, 0583291323, 665743, Jack Brown, 090592, 1
New client successfully created!
Please enter the number to the corresponding task you want to run.
3
Please enter your input in the following format: Email address input, Mailing address, Home phone number, Name
iysha@yahoo.com, 7482 Bucher Lane, 4029310423, 5029382311, 4029321132, 884721, Iysha Zimmerer, 0:
New client successfully created!
Please enter the number to the corresponding task you want to run.
3
Please enter your input in the following format: Email address input, Mailing address, Home phone number, Name
jeffery.g@yahoo.com, 592 Bumpy Lane, 4928310422, 9582031140, 4827391042, 993123, Jeffery Gonzalez, 0:
New client successfully created!
Please enter the number to the corresponding task you want to run.
3
Please enter your input in the following format: Email address input, Mailing address, Home phone number, Name
stacy_mockenz@outlook.com, 482 Horse Dr., 3928193922, 3819382291, 9482012234, 001942, Stacy Mockenz, 0:
New client successfully created!
Please enter the number to the corresponding task you want to run.

```

Please enter the number to the corresponding task you want to run.

3

Please enter your input in the following format: Email address input, Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB, Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external org. (Y or N), Join date, Training date, Training location, Team name, Is active (Y or N), Jan hours worked, Feb hours worked, March hours worked, April hours worked, May hours worked, June hours worked, July hours worked, Aug hours worked, Sept hours worked, Oct hours worked, Nov hours worked, Dec hours worked

jack@gmail.com, 764 Crest Dr., 0593021132, 0492310492, 0583291323, 665743, Jack Brown, 090592, White, M, CE, Y, N, 031207, 081417, Room 113, Team Brown, Y, 0, 0, 19, 29, 30, 10, 0, 17, 64, 0, 9, 15

```
New client successfully created!
Please enter the number to the corresponding task you want to run.
```

3

Please enter your input in the following format: Email address input, Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB, Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external org. (Y or N), Join date, Training date, Training location, Team name, Is active (Y or N), Jan hours worked, Feb hours worked, March hours worked, April hours worked, May hours worked, June hours worked, July hours worked, Aug hours worked, Sept hours worked, Oct hours worked, Nov hours worked, Dec hours worked

iysha@yahoo.com, 7482 Bucher Lane, 4029310423, 5029382311, 4029321132, 884721, Iysha Zimmerz, 071487, Indian, F, Art, N, N, 031207, 081417, Room 042, Team Gold, Y, 10, 64, 20, 48, 19, 0, 4, 18, 6, 9, 10, 90

```
New client successfully created!
Please enter the number to the corresponding task you want to run.
```

3

Please enter your input in the following format: Email address input, Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB, Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external

org. (Y or N), Join date, Training date, Training location, Team name, Is active (Y or N), Jan hours worked, Feb hours worked, March hours worked, April hours worked, May hours worked, June hours worked, July hours worked, Aug hours worked, Sept hours worked, Oct hours worked, Nov hours worked, Dec hours worked

jeffery.g@yahoo.com, 592 Bumpy Lane, 4928310422, 9582031140, 4827391042, 993123, Jeffery Gonzalez, 071797, Hispanic, M, Literature, Y, Y, 031207, 081417, Room LL23, Team White, Y, 95, 21, 0, 0, 0, 0, 71, 23, 19, 4, 82, 1

New client successfully created!

Please enter the number to the corresponding task you want to run.

3

Please enter your input in the following format: Email address input, Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB, Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external org. (Y or N), Join date, Training date, Training location, Team name, Is active (Y or N), Jan hours worked, Feb hours worked, March hours worked, April hours worked, May hours worked, June hours worked, July hours worked, Aug hours worked, Sept hours worked, Oct hours worked, Nov hours worked, Dec hours worked

stacy_mockenz@outlook.com, 482 Horse Dr., 3928193922, 3819382291, 9482012234, 001942, Stacy Mockenz, 051657, White, F, English, N, Y, 031207, 081417, Room BN382, Team Black, Y, 1, 0, 0, 0, 0, 0, 0, 0, 93, 17, 86

New client successfully created!

Please enter the number to the corresponding task you want to run.

3

Please enter your input in the following format: Email address input, Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB, Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external org. (Y or N), Join date, Training date, Training location, Team name, Is active (Y or N), Jan hours worked, Feb hours worked, March hours worked, April hours worked, May hours worked, June hours worked, July hours worked, Aug hours worked, Sept hours worked, Oct hours worked, Nov hours worked, Dec hours worked

thedominator@wow.net, 0192 Oak Ridge Rd., 3842831123, 8472931123, 5829381123, 112233, Don Black, 847566, African American, M, Cook, N, N, 031207, 081417, Room 948, Team Yellow, N, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

New client successfully created!

CS4513 x CONTACT_INFO x QUERY4 x

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL

Sort.. Filter:

| | EMAIL_ADDRESS | MAILING_ADDRESS | HOME_PHONE_NUMBER | WORK_PHONE_NUMBER | CELL_PHONE_NUMBER |
|----|---------------------------|----------------------|-------------------|-------------------|-------------------|
| 1 | emily@ou.edu | 842 Bobs Lane | 523921423 | 8574839284 | 293577382 |
| 2 | zack@ou.edu | 8431 West Drive | 5823258522 | 958424855 | 2031139482 |
| 3 | bill@gmail.edu | Three Partners Place | 459011922 | 492310422 | 942314232 |
| 4 | jerry.m.jones@hotmail.net | Five Partners Place | 492319322 | 942312332 | 492310442 |
| 5 | sarah-johnson571@msu.edu | 492 Palace Place | 492319332 | 5828329522 | 5842392333 |
| 6 | jack@gmail.com | 764 Crest Dr. | 593021132 | 492310492 | 583291323 |
| 7 | iysha@yahoo.com | 7482 Bucher Lane | 4029310423 | 5029382311 | 4029321132 |
| 8 | jeffery.g@yahoo.com | 592 Bumpy Lane | 4928310422 | 9582031140 | 4827391042 |
| 9 | stacy_mockenz@outlook.com | 482 Horse Dr. | 3928193922 | 3819382291 | 9482012234 |
| 10 | thedominator@wow.net | 0192 Oak Ridge Rd, | 3842831123 | 8472931123 | 5829381123 |

CS4513 x PERSON x QUERY3 x

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL

Sort.. Filter:

| | SSN | EMAIL_ADDRESS | NAME | DOB | RACE | GENDER | PROFESSION | ISONMAILINGLIST | ISAFFILIA |
|----|--------|----------------|---------------|-------|--------|--------|-------------|-----------------|-----------|
| 1 | 123456 | emily@ou.edu | Emily Black | 12... | White | F | CS | N | N |
| 2 | 744210 | zack@ou.edu | Zack Morenio | 91893 | White | M | EE | Y | N |
| 3 | 567890 | bill@gmail.edu | Bill Gardener | 10476 | Afr... | M | Physics | Y | Y |
| 4 | 987654 | jerry.m.jon... | Jerry Mich... | 62101 | Asian | M | Marketing | N | Y |
| 5 | 321098 | sarah-johns... | Sarah Johnson | 62101 | Mex... | F | Human Re... | N | N |
| 6 | 665743 | jack@gmail.com | Jack Brown | 90592 | White | M | CE | Y | N |
| 7 | 884721 | iysha@yahoo... | Iysha Zimmerz | 71487 | Indian | F | Art | N | N |
| 8 | 993123 | jeffery.g@y... | Jeffery Go... | 71797 | His... | M | Literature | Y | Y |
| 9 | 1942 | stacy_mocke... | Stacy Mockenz | 51657 | White | F | English | N | Y |
| 10 | 112233 | thedominato... | Don Black | 84... | Afr... | M | Cook | N | N |

| CS4513 x VOLUNTEER x QUERY3 x | | | | |
|--|--------|-----------|---------------|-------------------|
| Columns Data Model Constraints Grants Statistics Triggers Flashback Dependence | | | | |
| Sort.. Filter: | | | | |
| | SSN | JOIN_DATE | TRAINING_DATE | TRAINING_LOCATION |
| 1 | 665743 | 31207 | 81417 | Room 113 |
| 2 | 884721 | 31207 | 81417 | Room 042 |
| 3 | 993123 | 31207 | 81417 | Room LL23 |
| 4 | 1942 | 31207 | 81417 | Room BN382 |
| 5 | 112233 | 31207 | 81417 | Room 948 |

| CS4513 x SERVES_ON x QUERY3 x | | | | | | | | | | | | | | | | |
|---|--------|-------------|----------|------|------|------|------|------|------|------|------|------|------|------|------|--|
| Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL | | | | | | | | | | | | | | | | |
| Sort.. Filter: | | | | | | | | | | | | | | | | |
| | SSN | TEAM_NAME | ISACTIVE | J... | F... | M... | A... | M... | J... | J... | A... | S... | O... | N... | D... | |
| 1 | 665743 | Team Brown | Y | 0 | 0 | 19 | 29 | 30 | 10 | 0 | 17 | 64 | 0 | 9 | 15 | |
| 2 | 884721 | Team Gold | Y | 10 | 64 | 20 | 48 | 19 | 0 | 4 | 18 | 6 | 9 | 10 | 90 | |
| 3 | 993123 | Team White | Y | 95 | 21 | 0 | 0 | 0 | 0 | 71 | 23 | 19 | 4 | 82 | 1 | |
| 4 | 1942 | Team Black | Y | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 93 | 17 | 86 | |
| 5 | 112233 | Team Yellow | N | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

6.4. Script file showing the testing of query 4

```

Problems @ Javadoc Declaration Console x
IP_JAVA_BLACK_EMILY [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Nov 13, 2016)

Please enter the number to the corresponding task you want to run.
4
Please enter your input in the following format: SSN, Jan hours worked, Feb
665743, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0
New hours a volunteer worked successfully created!
Please enter the number to the corresponding task you want to run.
4
Please enter your input in the following format: SSN, Jan hours worked, Feb
884721, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
New hours a volunteer worked successfully created!
Please enter the number to the corresponding task you want to run.
4
Please enter your input in the following format: SSN, Jan hours worked, Feb
993123, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0
New hours a volunteer worked successfully created!
Please enter the number to the corresponding task you want to run.
4

```

Please enter the number to the corresponding task you want to run.

4

Please enter your input in the following format: SSN, Jan hours worked, Feb hours worked, March hours worked, April hours worked, May hours worked, June hours worked, July hours worked, Aug hours worked, Sept hours worked, Oct hours worked, Nov hours worked, Dec hours worked

665743, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0

New hours a volunteer worked successfully created!

Please enter the number to the corresponding task you want to run.

4

Please enter your input in the following format: SSN, Jan hours worked, Feb hours worked, March hours worked, April hours worked, May hours worked, June hours worked, July hours worked, Aug hours worked, Sept hours worked, Oct hours worked, Nov hours worked, Dec hours worked

884721, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

New hours a volunteer worked successfully created!

Please enter the number to the corresponding task you want to run.

4

Please enter your input in the following format: SSN, Jan hours worked, Feb hours worked, March hours worked, April hours worked, May hours worked, June hours worked, July hours worked, Aug hours worked, Sept hours worked, Oct hours worked, Nov hours worked, Dec hours worked

993123, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0

New hours a volunteer worked successfully created!

Please enter the number to the corresponding task you want to run.

4

Please enter your input in the following format: SSN, Jan hours worked, Feb hours worked, March hours worked, April hours worked, May hours worked, June hours worked, July hours worked, Aug hours worked, Sept hours worked, Oct hours worked, Nov hours worked, Dec hours worked

001942, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

New hours a volunteer worked successfully created!

Please enter the number to the corresponding task you want to run.

4

Please enter your input in the following format: SSN, Jan hours worked, Feb hours worked, March hours worked, April hours worked, May hours worked, June hours worked, July hours worked, Aug hours worked, Sept hours worked, Oct hours worked, Nov hours worked, Dec hours worked

112233, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0

New hours a volunteer worked successfully created!

| | SSN | TEAM_NAME | ISACTIVE | J... | F... | M... | A... | M... | J... | J... | A... | S... | O... | N... | D... |
|---|--------|-------------|----------|------|------|------|------|------|------|------|------|------|------|------|------|
| 1 | 665743 | Team Brown | Y | 0 | 0 | 20 | 29 | 30 | 10 | 0 | 17 | 64 | 0 | 9 | 15 |
| 2 | 884721 | Team Gold | Y | 11 | 64 | 20 | 48 | 19 | 0 | 4 | 18 | 6 | 9 | 10 | 90 |
| 3 | 993123 | Team White | Y | 95 | 21 | 0 | 0 | 1 | 0 | 71 | 23 | 19 | 4 | 82 | 1 |
| 4 | 1942 | Team Black | Y | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 93 | 17 | 86 |
| 5 | 112233 | Team Yellow | N | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

6.5. Script file showing the testing of query 5

```

IP_JAVA_BLACK_EMILY [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Nov 13, 2017, 7:28:52 PM)
(19) Export: Retrieve names and mailing addresses of all people on the mailing list and outp
(20) Quit.

Please enter the number to the corresponding task you want to run.
5
Please enter your input in the following format: Email address, Mailing address, Home phone
jimmy-b@yahoo.com, 623 Crystal Cr., 7482393122, 5827321122, 5843921123, 339901, Jimmy Barker
New employee successfully created!
Please enter the number to the corresponding task you want to run.
5
Please enter your input in the following format: Email address, Mailing address, Home phone
lennard423@yahoo.com, 634 Banana Lane, 8472912235, 8472931123, 5827301123, 564738, Lennard K
New employee successfully created!
Please enter the number to the corresponding task you want to run.
5
Please enter your input in the following format: Email address, Mailing address, Home phone
billyw@yahoo.com, 231 East Coast Drive, 4827118233, 5827371123, 5827321123, 163296, Billy Wi
New employee successfully created!
Please enter the number to the corresponding task you want to run.

```

Please enter the number to the corresponding task you want to run.

5
Please enter your input in the following format: Email address, Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB (Ex: 121495 = Dec. 14th, 1995), Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external organization (Y or N), Salary, Martial status, Hire date, Team name, Date of report (Ex: 121495 = Dec. 14th, 1995), Description
yyija@yahoo.com, 8492 Black Drive, 9382939113, 827382738, 92838247, 998877, Yyija Mathias, 040383, African American, F, CS, N, N, 56990, Single, 091899, Team Gold, 121311, Went over costs of project
New employee successfully created!
Please enter the number to the corresponding task you want to run.

5
Please enter your input in the following format: Email address, Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB (Ex: 121495 = Dec. 14th, 1995), Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external organization (Y or N), Salary, Martial status, Hire date, Team name, Date of report (Ex: 121495 = Dec. 14th, 1995), Description
joshk@yahoo.com, 95834 Blue Creek Lane, 4827472233, 9482731123, 4823928193, 645221, Josh Kipps, 051857, White, M, EE, Y, N, 95000, Divorced, 092896, Team Yellow, 050607, Discussed future plans with project
New employee successfully created!
Please enter the number to the corresponding task you want to run.

5
Please enter your input in the following format: Email address, Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB (Ex: 121495 = Dec. 14th, 1995), Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external organization (Y or N), Salary, Martial status, Hire date, Team name, Date of report (Ex: 121495 = Dec. 14th, 1995), Description

jimmy-b@yahoo.com, 623 Crystal Cr., 7482393122, 5827321122, 5843921123, 339901, Jimmy Barker, 112369, White, M, ME, Y, Y, 43525, Married, 091472, Team Brown, 071316, Laid out plans for new project

New employee successfully created!

Please enter the number to the corresponding task you want to run.

5

Please enter your input in the following format: Email address, Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB (Ex: 121495 = Dec. 14th, 1995), Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external organization (Y or N), Salary, Martial status, Hire date, Team name, Date of report (Ex: 121495 = Dec. 14th, 1995), Description

lennard423@yahoo.com, 634 Banana Lane, 8472912235, 8472931123, 5827301123, 564738, Lennard Kennard, 040567, Asian, M, ME/CE, N, Y, 102000, Widow, 111389, Team Black, 121311, Discussed new budget cuts and how it impacts the team

New employee successfully created!

Please enter the number to the corresponding task you want to run.

5

Please enter your input in the following format: Email address, Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB (Ex: 121495 = Dec. 14th, 1995), Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external organization (Y or N), Salary, Martial status, Hire date, Team name, Date of report (Ex: 121495 = Dec. 14th, 1995), Description

billyw@yahoo.com, 231 East Coast Drive, 4827118233, 5827371123, 5827321123, 163296, Billy Willyard, 071492, White, M, CS/CE, N, N, 32000, Single, 081109, Team White, 110916, Discussed and implemented new structure for backend system

New employee successfully created!

| CS4513 x QUERY5 x CONTACT_INFO x | | | | | |
|---|---------------------------|-----------------------|-------------------|-------------------|-------------------|
| Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL | | | | | |
| Sort.. Filter: | | | | | |
| | EMAIL_ADDRESS | MAILING_ADDRESS | HOME_PHONE_NUMBER | WORK_PHONE_NUMBER | CELL_PHONE_NUMBER |
| 1 | emily@ou.edu | 842 Bobs Lane | 523921423 | 8574839284 | 293577382 |
| 2 | zack@ou.edu | 8431 West Drive | 5823258522 | 958424855 | 2031139482 |
| 3 | bill@gmail.edu | Three Partners Place | 459011922 | 492310422 | 942314232 |
| 4 | jerry.m.jones@hotmail.net | Five Partners Place | 492319322 | 942312332 | 492310442 |
| 5 | sarah-johnson571@msu.edu | 492 Palace Place | 492319332 | 5828329522 | 5842392333 |
| 6 | jack@gmail.com | 764 Crest Dr. | 593021132 | 492310492 | 583291323 |
| 7 | iysha@yahoo.com | 7482 Bucher Lane | 4029310423 | 5029382311 | 4029321132 |
| 8 | jeffery.g@yahoo.com | 592 Bumpy Lane | 4928310422 | 9582031140 | 4827391042 |
| 9 | stacy_mockenz@outlook.com | 482 Horse Dr. | 3928193922 | 3819382291 | 9482012234 |
| 10 | thedominator@wow.net | 0192 Oak Ridge Rd, | 3842831123 | 8472931123 | 5829381123 |
| 11 | yyija@yahoo.com | 8492 Black Drive | 9382939113 | 827382738 | 92838247 |
| 12 | joshk@yahoo.com | 95834 Blue Creek Lane | 4827472233 | 9482731123 | 4823928193 |
| 13 | jimmy-b@yahoo.com | 623 Crystal Cr. | 7482393122 | 5827321122 | 5843921123 |
| 14 | lennard423@yahoo.com | 634 Banana Lane | 8472912235 | 8472931123 | 5827301123 |
| 15 | billyw@yahoo.com | 231 East Coast Drive | 4827118233 | 5827371123 | 5827321123 |

| CS4513 x QUERY5 x PERSON x | | | | | | | | |
|---|--------|----------------|---------------|-------|--------|--------|-------------|-----------------|
| Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL | | | | | | | | |
| Sort.. Filter: | | | | | | | | |
| | SSN | EMAIL_ADDRESS | NAME | DOB | RACE | GENDER | PROFESSION | ISONMAILINGLIST |
| 1 | 123456 | emily@ou.edu | Emily Black | 12... | White | F | CS | N |
| 2 | 744210 | zack@ou.edu | Zack Morenio | 91893 | White | M | EE | Y |
| 3 | 567890 | bill@gmail.edu | Bill Gardener | 10476 | Afr... | M | Physics | Y |
| 4 | 987654 | jerry.m.jon... | Jerry Mich... | 62101 | Asian | M | Marketing | N |
| 5 | 321098 | sarah-johns... | Sarah Johnson | 62101 | Mex... | F | Human Re... | N |
| 6 | 665743 | jack@gmail.com | Jack Brown | 90592 | White | M | CE | Y |
| 7 | 884721 | iysha@yahoo... | Iysha Zimmerz | 71487 | Indian | F | Art | N |
| 8 | 993123 | jeffery.g@y... | Jeffery Go... | 71797 | His... | M | Literature | Y |
| 9 | 1942 | stacy_mocke... | Stacy Mockenz | 51657 | White | F | English | N |
| 10 | 112233 | thedominato... | Don Black | 84... | Afr... | M | Cook | N |
| 11 | 998877 | yyija@yahoo... | Yyija Mathias | 40383 | Afr... | F | CS | N |
| 12 | 645221 | joshk@yahoo... | Josh Kipps | 51857 | White | M | EE | Y |
| 13 | 339901 | jimmy-b@yah... | Jimmy Barker | 11... | White | M | ME | Y |
| 14 | 564738 | lennard423@... | Lennard Ke... | 40567 | Asian | M | ME/CE | N |
| 15 | 163296 | billyw@yaho... | Billy Will... | 71492 | White | M | CS/CE | N |

CS4513 x QUERY5 x EMPLOYEE x

Columns Data Model Constraints Grants Statistics Triggers Flashback

Sort.. Filter:

| | SSN | SALARY | MARTIAL_STATUS | HIRE_DATE |
|---|--------|--------|----------------|-----------|
| 1 | 998877 | 56990 | Single | 91899 |
| 2 | 645221 | 95000 | Divorced | 92896 |
| 3 | 339901 | 43525 | Married | 91472 |
| 4 | 564738 | 102000 | Widow | 111389 |
| 5 | 163296 | 32000 | Single | 81109 |

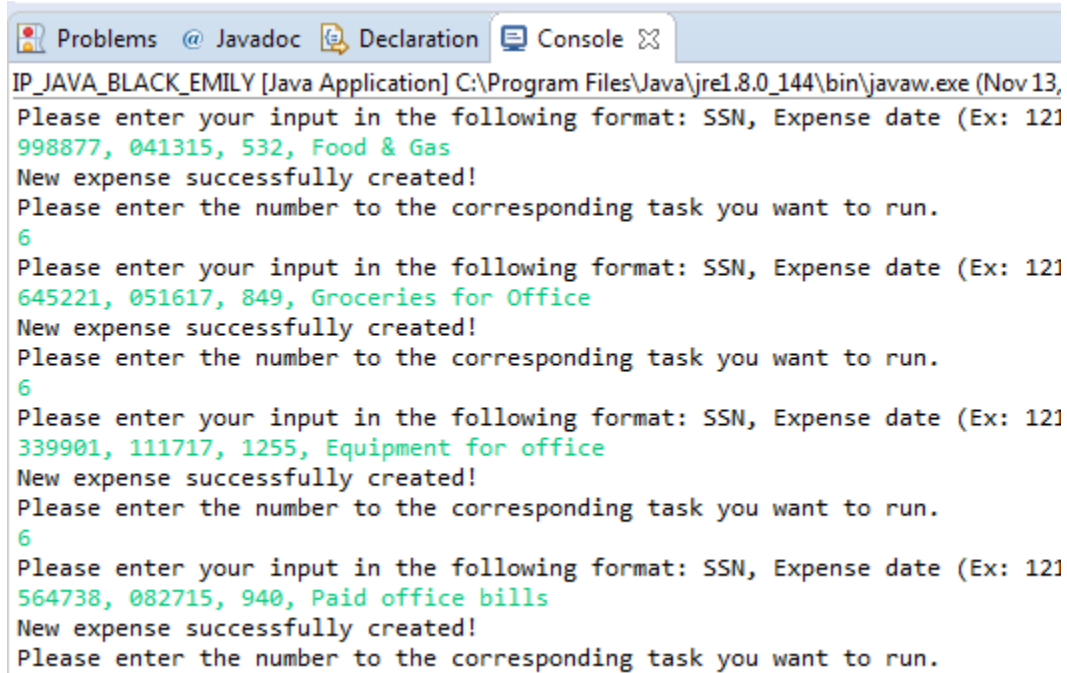
CS4513 x REPORTS_TO x

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL

Sort.. Filter:

| | TEAM_NAME | SSN | DATE_OF_REPORT | DESCRIPTION |
|----|-------------|--------|----------------|--|
| 1 | Team Brown | 339901 | 71316 | Laid out plans for new project |
| 2 | Team Black | 564738 | 121311 | Discussed new budget cuts and how it impacts the team |
| 3 | Team White | 163296 | 110916 | Discussed and implemented new structure for backend system |
| 4 | Team Gold | 998877 | 121311 | Went over costs of project |
| 5 | Team Yellow | 645221 | 50607 | Discussed future plans with project |
| 6 | Team Black | 564738 | 111516 | Went over analytics of group project |
| 7 | Team Black | 564738 | 121309 | Discussed what to have for lunch next week |
| 8 | Team Gold | 163296 | 91514 | Discussed possible party after completion of project |
| 9 | Team White | 645221 | 21713 | Went of graphs of the team's progress |
| 10 | Team Yellow | 998877 | 52215 | Discussed upcoming features for project |
| 11 | Team Brown | 998877 | 62917 | Laid out new features for existing project |

6.6. Script file showing the testing of query 6



```

IP_JAVA_BLACK_EMILY [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Nov 13, 2015)
Please enter your input in the following format: SSN, Expense date (Ex: 121
998877, 041315, 532, Food & Gas
New expense successfully created!
Please enter the number to the corresponding task you want to run.
6
Please enter your input in the following format: SSN, Expense date (Ex: 121
645221, 051617, 849, Groceries for Office
New expense successfully created!
Please enter the number to the corresponding task you want to run.
6
Please enter your input in the following format: SSN, Expense date (Ex: 121
339901, 111717, 1255, Equipment for office
New expense successfully created!
Please enter the number to the corresponding task you want to run.
6
Please enter your input in the following format: SSN, Expense date (Ex: 121
564738, 082715, 940, Paid office bills
New expense successfully created!
Please enter the number to the corresponding task you want to run.

```

Please enter the number to the corresponding task you want to run.

6

Please enter your input in the following format: SSN, Expense date (Ex: 121495 = Dec. 14th, 1995), Expense amount, Expense description

998877, 041315, 532, Food & Gas

New expense successfully created!

Please enter the number to the corresponding task you want to run.

6

Please enter your input in the following format: SSN, Expense date (Ex: 121495 = Dec. 14th, 1995), Expense amount, Expense description

645221, 051617, 849, Groceries for Office

New expense successfully created!

Please enter the number to the corresponding task you want to run.

6

Please enter your input in the following format: SSN, Expense date (Ex: 121495 = Dec. 14th, 1995), Expense amount, Expense description

339901, 111717, 1255, Equipment for office

New expense successfully created!

Please enter the number to the corresponding task you want to run.

6

Please enter your input in the following format: SSN, Expense date (Ex: 121495 = Dec. 14th, 1995), Expense amount, Expense description

564738, 082715, 940, Paid office bills

New expense successfully created!

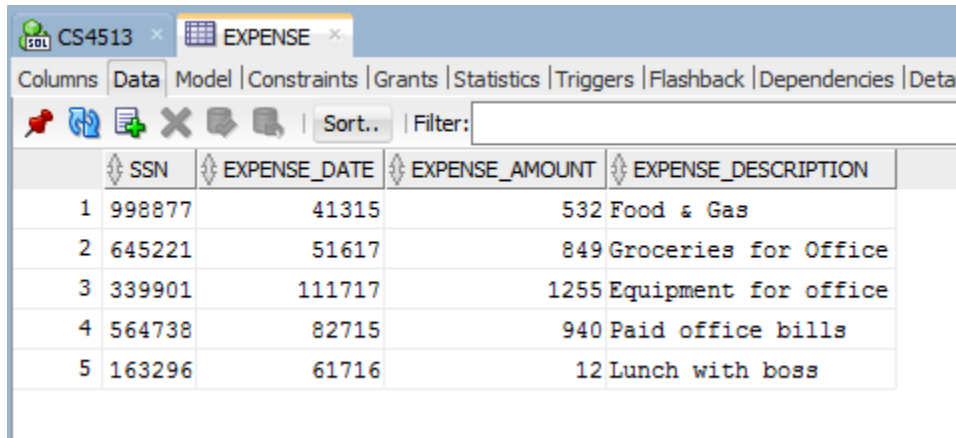
Please enter the number to the corresponding task you want to run.

6

Please enter your input in the following format: SSN, Expense date (Ex: 121495 = Dec. 14th, 1995), Expense amount, Expense description

163296, 061716, 12, Lunch with boss

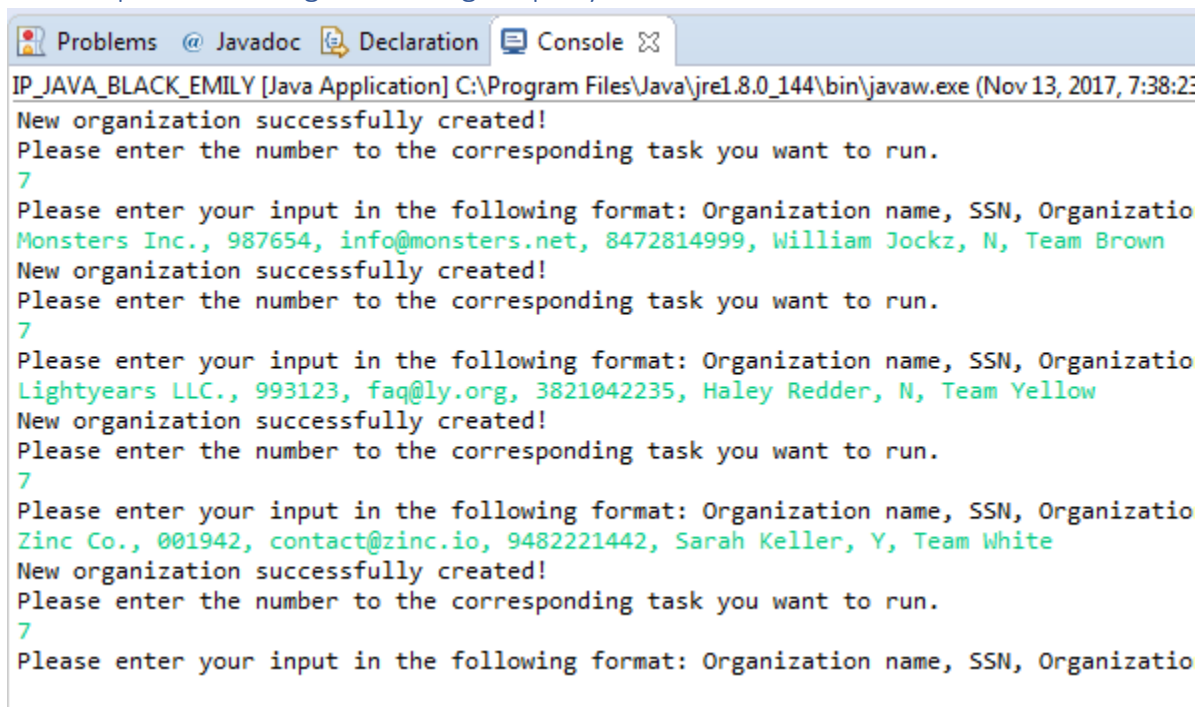
New expense successfully created!



The screenshot shows the SQL Server Enterprise Manager interface with the 'EXPENSE' table selected. The table has four columns: SSN, EXPENSE_DATE, EXPENSE_AMOUNT, and EXPENSE_DESCRIPTION. There are five rows of data displayed.

| | SSN | EXPENSE_DATE | EXPENSE_AMOUNT | EXPENSE_DESCRIPTION |
|---|--------|--------------|----------------|----------------------|
| 1 | 998877 | 41315 | 532 | Food & Gas |
| 2 | 645221 | 51617 | 849 | Groceries for Office |
| 3 | 339901 | 111717 | 1255 | Equipment for office |
| 4 | 564738 | 82715 | 940 | Paid office bills |
| 5 | 163296 | 61716 | 12 | Lunch with boss |

6.7. Script file showing the testing of query 7



The screenshot shows the console output of a Java application. The application prompts the user to enter a task number (7) and then provides a format for organization data. It successfully creates four organizations: Monsters Inc., Lightyears LLC., Zinc Co., and Sarkz Inc., each with a unique SSN and team name.

```

IP_JAVA_BLACK_EMILY [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Nov 13, 2017, 7:38:22)
New organization successfully created!
Please enter the number to the corresponding task you want to run.
7
Please enter your input in the following format: Organization name, SSN, Organization
Monsters Inc., 987654, info@monsters.net, 8472814999, William Jockz, N, Team Brown
New organization successfully created!
Please enter the number to the corresponding task you want to run.
7
Please enter your input in the following format: Organization name, SSN, Organization
Lightyears LLC., 993123, faq@ly.org, 3821042235, Haley Redder, N, Team Yellow
New organization successfully created!
Please enter the number to the corresponding task you want to run.
7
Please enter your input in the following format: Organization name, SSN, Organization
Zinc Co., 001942, contact@zinc.io, 9482221442, Sarah Keller, Y, Team White
New organization successfully created!
Please enter the number to the corresponding task you want to run.
7
Please enter your input in the following format: Organization name, SSN, Organization

```

Please enter the number to the corresponding task you want to run.

7

Please enter your input in the following format: Organization name, SSN, Organization mailing address, Organization phone number, Organization contact person, Is org anonymous (Y or N), Team name

Sarkz Inc., 567890, 941 Apple Dr., 9482339123, Willis Zackerman, Y, Team Gold

New organization successfully created!

Please enter the number to the corresponding task you want to run.

7

Please enter your input in the following format: Organization name, SSN, Organization mailing address, Organization phone number, Organization contact person, Is org anonymous (Y or N), Team name

Monsters Inc., 987654, 942 Apple Dr., 8472814999, William Jockz, N, Team Brown

New organization successfully created!

Please enter the number to the corresponding task you want to run.

7

Please enter your input in the following format: Organization name, SSN, Organization mailing address, Organization phone number, Organization contact person, Is org anonymous (Y or N), Team name

Lightyears LLC., 993123, 8470 Skynet Dr., 3821042235, Haley Redder, N, Team Yellow

New organization successfully created!

Please enter the number to the corresponding task you want to run.

7

Please enter your input in the following format: Organization name, SSN, Organization mailing address, Organization phone number, Organization contact person, Is org anonymous (Y or N), Team name

Zinc Co., 001942, 8471 Skynet Dr., 9482221442, Sarah Keller, Y, Team White

New organization successfully created!

Please enter the number to the corresponding task you want to run.

7

Please enter your input in the following format: Organization name, SSN, Organization mailing address, Organization phone number, Organization contact person, Is org anonymous (Y or N), Team name







Blizzard Entertainment, 339901, 8472 Skynet Dr., 3528184421, Josh Yanzak, N, Team Black

New organization successfully created!

CS4513

EXTERNAL_ORGANIZATION

Columns | Data | Model | Constraints | Grants | Statistics | Triggers | Flashback | Dependencies | Details | Partitions | Indexes | SQL



Sort..

Filter:

| | ORGANIZATION_NAME | SSN | ORG_MAILING_ADDRESS | ORG_PHONE_NUMBER | ORG_CONTACT_PERSON | ISC |
|---|------------------------|--------|---------------------|------------------|--------------------|-----|
| 1 | Sarkz Inc. | 567890 | 941 Apple Dr. | 9482339123 | Willis Zackerman | Y |
| 2 | Monsters Inc. | 987654 | 942 Apple Dr. | 8472814999 | William Jockz | N |
| 3 | Lightyears LLC. | 993123 | 8470 Skynet Dr. | 3821042235 | Haley Redder | N |
| 4 | Zinc Co. | 1942 | 8471 Skynet Dr. | 9482221442 | Sarah Keller | Y |
| 5 | Blizzard Entertainment | 339901 | 8472 Skynet Dr. | 3528184421 | Josh Yanzak | N |

| CS4513 | | MAY_SPONSOR | | QUERY7 | | | | | |
|---|------------------------|-------------|--|--------|--|--|--|--|--|
| Columns Data Model Constraints Grants Statistics Triggers | | | | | | | | | |
| Sort.. Filter: | | | | | | | | | |
| | ORGANIZATION_NAME | TEAM_NAME | | | | | | | |
| 1 | Sarkz Inc. | Team Gold | | | | | | | |
| 2 | Monsters Inc. | Team Brown | | | | | | | |
| 3 | Lightyears LLC. | Team Yellow | | | | | | | |
| 4 | Zinc Co. | Team White | | | | | | | |
| 5 | Blizzard Entertainment | Team Black | | | | | | | |

6.8. Script file showing the testing of query 8

```

IP JAVA_BLACK_EMILY [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Nov 13, 2017, 7:47:20 PM)
Please enter your input in the following format: Email address, Mailing address, Home ph
yanzak@hotmail.com, 726 Riverwind Dr., 2847284421, 7562839123, 8265821242, 238901, Jill
New donor successfully created!
Please enter the number to the corresponding task you want to run.
8
Please enter your input in the following format: Email address, Mailing address, Home ph
sarah-h31@hotmail.com, 734 Knights Lane, 4728194422, 4829548233, 8572948111, 887201, Sar
New donor successfully created!
Please enter the number to the corresponding task you want to run.
8
Please enter your input in the following format: Email address, Mailing address, Home ph
847291@ou.edu, 9278 Tall Tree Way, 5847382234, 5837284432, 5837432212, 772056, Buddy Est
New donor successfully created!
Please enter the number to the corresponding task you want to run.
8
Please enter your input in the following format: Email address, Mailing address, Home ph
hillard@hotmail.com, 113 Complex Road, 6954637222, 5728375922, 8472334124, 768301, Hilla
New donor successfully created!
Please enter the number to the corresponding task you want to run.

```

Please enter the number to the corresponding task you want to run.

8

Please enter your input in the following format: Email address, Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB (Ex: 121495 = Dec. 14th, 1995), Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external organization (Y or N), Is donor anonymous (Y or N)

yanzak@hotmail.com, 726 Riverwind Dr., 2847284421, 7562839123, 8265821242, 238901, Jill Yanzak, 052399, White, F, EE, Y, N, Y

New donor successfully created!

Please enter the number to the corresponding task you want to run.

8

Please enter your input in the following format: Email address, Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB (Ex: 121495 = Dec. 14th, 1995), Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external organization (Y or N), Is donor anonymous (Y or N)

sarah-h31@hotmail.com, 734 Knights Lane, 4728194422, 4829548233, 8572948111, 887201, Sarah Haah, 091295, Mexican, F, CCE, N, Y, Y

New donor successfully created!

Please enter the number to the corresponding task you want to run.

8

Please enter your input in the following format: Email address, Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB (Ex: 121495 = Dec. 14th, 1995), Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external organization (Y or N), Is donor anonymous (Y or N)

847291@ou.edu, 9278 Tall Tree Way, 5847382234, 5837284432, 5837432212, 772056, Buddy Esther, 031299, African American, F, Biomedical, N, Y, N

New donor successfully created!

Please enter the number to the corresponding task you want to run.

8

Please enter your input in the following format: Email address, Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB (Ex: 121495 = Dec. 14th, 1995), Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external organization (Y or N), Is donor anonymous (Y or N)

hillard@hotmail.com, 113 Complex Road, 6954637222, 5728375922, 8472334124, 768301, Hillard Keya, 012797, African American, M, CS/EE, Y, Y, Y

New donor successfully created!

Please enter the number to the corresponding task you want to run.

8

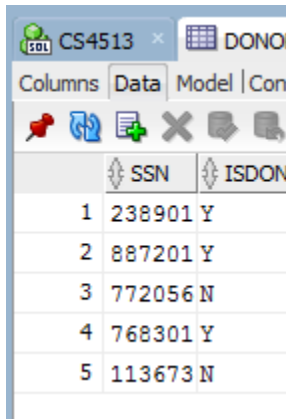
Please enter your input in the following format: Email address, Mailing address, Home phone number, Work phone number, Cell phone number, SSN, Name, DoB (Ex: 121495 = Dec. 14th, 1995), Race, Gender, Profession, Is on mailing list (Y or N), Is affiliated with external organization (Y or N), Is donor anonymous (Y or N)

133tkid090@hotmail.com, 847 Black Lane, 5736283312, 4827341124, 5837231223, 113673, Mathew Joller, 122699, White, M, Business, N, Y, N

New donor successfully created!

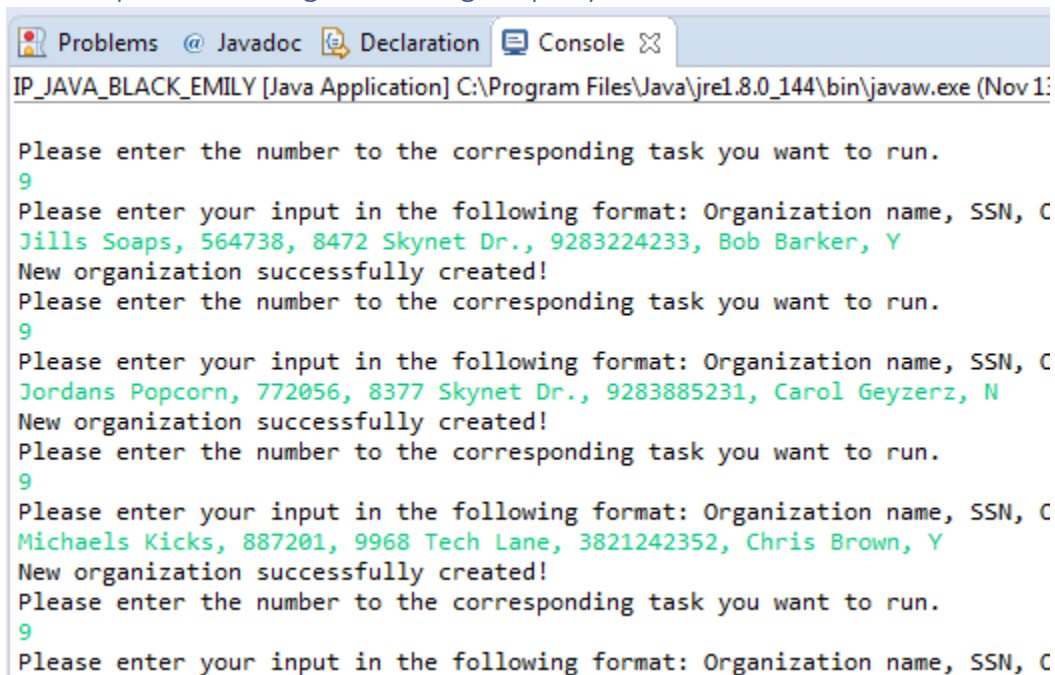
| CS4513 x CONTACT_INFO x QUERY8 x | | | | | |
|---|---------------------------|-----------------------|-------------------|-------------------|-------------------|
| Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL | | | | | |
| Sort.. Filter: | | | | | |
| | EMAIL_ADDRESS | MAILING_ADDRESS | HOME_PHONE_NUMBER | WORK_PHONE_NUMBER | CELL_PHONE_NUMBER |
| 1 | emily@ou.edu | 842 Bobs Lane | 523921423 | 8574839284 | 293577382 |
| 2 | zack@ou.edu | 8431 West Drive | 5823258522 | 958424855 | 2031139482 |
| 3 | bill@gmail.edu | Three Partners Place | 459011922 | 492310422 | 942314232 |
| 4 | jerry.m.jones@hotmail.net | Five Partners Place | 492319322 | 942312332 | 492310442 |
| 5 | sarah-johnson571@msu.edu | 492 Palace Place | 492319332 | 5828329522 | 5842392333 |
| 6 | jack@gmail.com | 764 Crest Dr. | 593021132 | 492310492 | 583291323 |
| 7 | iysha@yahoo.com | 7482 Bucher Lane | 4029310423 | 5029382311 | 4029321132 |
| 8 | jeffery.g@yahoo.com | 592 Bumpy Lane | 4928310422 | 9582031140 | 4827391042 |
| 9 | stacy_mockenz@outlook.com | 482 Horse Dr. | 3928193922 | 3819382291 | 9482012234 |
| 10 | thedominator@wow.net | 0192 Oak Ridge Rd, | 3842831123 | 8472931123 | 5829381123 |
| 11 | yyija@yahoo.com | 8492 Black Drive | 9382939113 | 827382738 | 92838247 |
| 12 | joshk@yahoo.com | 95834 Blue Creek Lane | 4827472233 | 9482731123 | 4829328193 |
| 13 | jimmy-b@yahoo.com | 623 Crystal Cr. | 7482393122 | 5827321122 | 5843921123 |
| 14 | lennard423@yahoo.com | 634 Banana Lane | 8472912235 | 8472931123 | 5827301123 |
| 15 | billyw@yahoo.com | 231 East Coast Drive | 4827118233 | 5827371123 | 5827321123 |
| 16 | yanzak@hotmail.com | 726 Riverwind Dr. | 2847284421 | 7562839123 | 8265821242 |
| 17 | sarah-h31@hotmail.com | 734 Knights Lane | 4728194422 | 4829548233 | 8572948111 |
| 18 | 847291@ou.edu | 9278 Tall Tree Way | 5847382234 | 5837284432 | 5837432212 |
| 19 | hillard@hotmail.com | 113 Complex Road | 6954637222 | 5728375922 | 8472334124 |
| 20 | 133tkid090@hotmail.com | 847 Black Lane | 5736283312 | 4827341124 | 5837231223 |

| CS4513 x PERSON x | | | | | | | | |
|---|--------|----------------|---------------|-------|--------|--------|-------------|-----------------|
| Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL | | | | | | | | |
| Sort.. Filter: | | | | | | | | |
| | SSN | EMAIL_ADDRESS | NAME | DOB | RACE | GENDER | PROFESSION | ISONMAILINGLIST |
| 1 | 123456 | emily@ou.edu | Emily Black | 12... | White | F | CS | N |
| 2 | 744210 | zack@ou.edu | Zack Morenio | 91893 | White | M | EE | Y |
| 3 | 567890 | bill@gmail.edu | Bill Gardener | 10476 | Afr... | M | Physics | Y |
| 4 | 987654 | jerry.m.jon... | Jerry Mich... | 62101 | Asian | M | Marketing | N |
| 5 | 321098 | sarah-johns... | Sarah Johnson | 62101 | Mex... | F | Human Re... | N |
| 6 | 665743 | jack@gmail.com | Jack Brown | 90592 | White | M | CE | Y |
| 7 | 884721 | iysha@yahoo... | Iysha Zimmerz | 71487 | Indian | F | Art | N |
| 8 | 993123 | jeffery.g@y... | Jeffery Go... | 71797 | His... | M | Literature | Y |
| 9 | 1942 | stacy_mocke... | Stacy Mockenz | 51657 | White | F | English | N |
| 10 | 112233 | thedominato... | Don Black | 84... | Afr... | M | Cook | N |
| 11 | 998877 | yyija@yahoo... | Yyija Mathias | 40383 | Afr... | F | CS | N |
| 12 | 645221 | joshk@yahoo... | Josh Kipps | 51857 | White | M | EE | Y |
| 13 | 339901 | jimmy-b@yah... | Jimmy Barker | 11... | White | M | ME | Y |
| 14 | 564738 | lennard423@... | Lennard Ke... | 40567 | Asian | M | ME/CE | N |
| 15 | 163296 | billyw@yaho... | Billy Will... | 71492 | White | M | CS/CE | N |
| 16 | 772056 | 847291@ou.edu | Buddy Esther | 31299 | Afr... | F | Biomedical | N |
| 17 | 238901 | yanzak@hotm... | Jill Yanzak | 52399 | White | F | EE | Y |
| 18 | 887201 | sarah-h31@h... | Sarah Haah | 91295 | Mex... | F | CCE | N |
| 19 | 768301 | hillard@hot... | Hillard Keya | 12797 | Afr... | M | CS/EE | Y |
| 20 | 113673 | 133tkid090@... | Mathew Joller | 12... | White | M | Business | N |



| | SSN | ISDON |
|---|--------|-------|
| 1 | 238901 | Y |
| 2 | 887201 | Y |
| 3 | 772056 | N |
| 4 | 768301 | Y |
| 5 | 113673 | N |

6.9. Script file showing the testing of query 9



```

Problems @ Javadoc Declaration Console
IP_JAVA_BLACK_EMILY [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Nov 1:

Please enter the number to the corresponding task you want to run.
9
Please enter your input in the following format: Organization name, SSN, C
Jills Soaps, 564738, 8472 Skynet Dr., 9283224233, Bob Barker, Y
New organization successfully created!
Please enter the number to the corresponding task you want to run.
9
Please enter your input in the following format: Organization name, SSN, C
Jordans Popcorn, 772056, 8377 Skynet Dr., 9283885231, Carol Geyzerz, N
New organization successfully created!
Please enter the number to the corresponding task you want to run.
9
Please enter your input in the following format: Organization name, SSN, C
Michaels Kicks, 887201, 9968 Tech Lane, 3821242352, Chris Brown, Y
New organization successfully created!
Please enter the number to the corresponding task you want to run.
9
Please enter your input in the following format: Organization name, SSN, C

```

Please enter the number to the corresponding task you want to run.

9

Please enter your input in the following format: Organization name, SSN, Organization mailing address, Organization phone number, Organization contact person, Is org anonymous (Y or N)

Jills Soaps, 564738, 8472 Skynet Dr., 9283224233, Bob Barker, Y

New organization successfully created!

Please enter the number to the corresponding task you want to run.

9

Please enter your input in the following format: Organization name, SSN, Organization mailing address, Organization phone number, Organization contact person, Is org anonymous (Y or N)

Jordans Popcorn, 772056, 8377 Skynet Dr., 9283885231, Carol Geyzerz, N

New organization successfully created!

Please enter the number to the corresponding task you want to run.

9

Please enter your input in the following format: Organization name, SSN, Organization mailing address, Organization phone number, Organization contact person, Is org anonymous (Y or N)

Michaels Kicks, 887201, 9968 Tech Lane, 3821242352, Chris Brown, Y

New organization successfully created!

Please enter the number to the corresponding task you want to run.

9

Please enter your input in the following format: Organization name, SSN, Organization mailing address, Organization phone number, Organization contact person, Is org anonymous (Y or N)

Michael Kors, 768301, 844 Gucci Lane, 9472385521, Kim Laurel, N

New organization successfully created!

Please enter the number to the corresponding task you want to run.

9

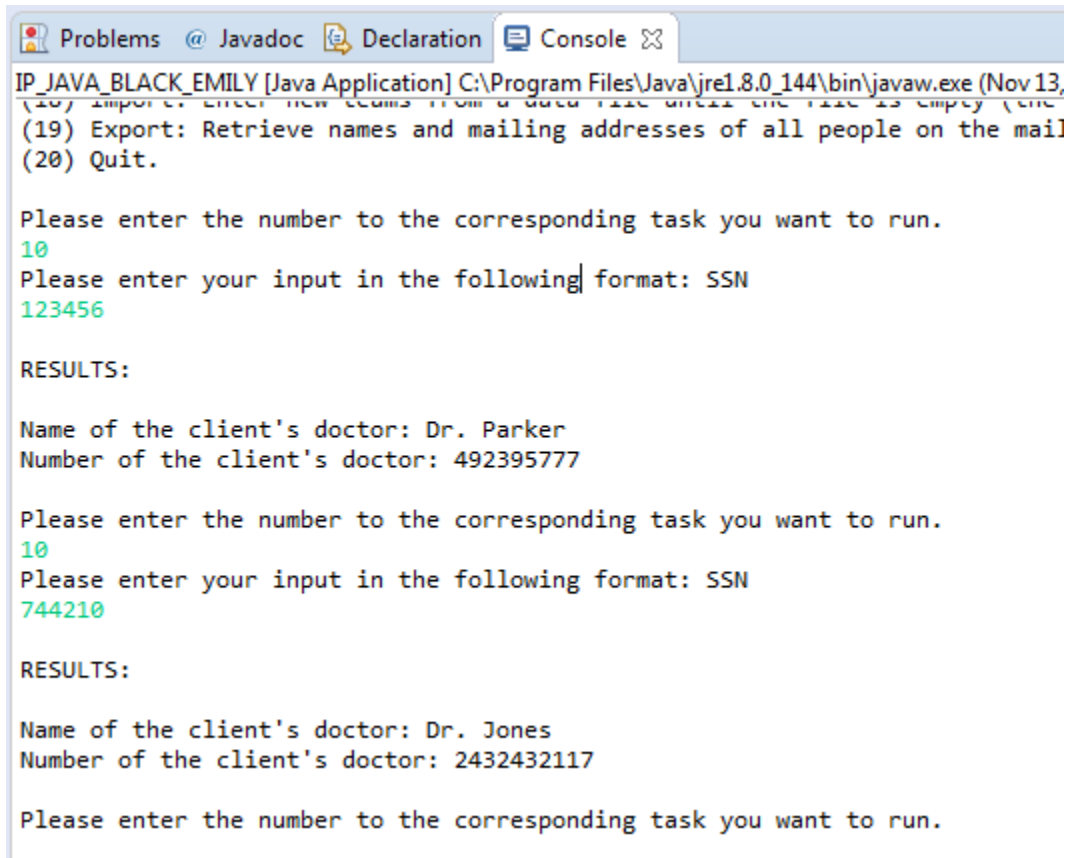
Please enter your input in the following format: Organization name, SSN, Organization mailing address, Organization phone number, Organization contact person, Is org anonymous (Y or N)

Nike Inc., 113673, 123 Swoosh Rd., 8572837712, David McDavis, N

New organization successfully created!

| EXTERNAL_ORGANIZATION | | | | | |
|--------------------------|-------------|---------------------|------------------|--------------------|------------------|
| ORGANIZATION_NAME | SSN | ORG_MAILING_ADDRESS | ORG_PHONE_NUMBER | ORG_CONTACT_PERSON | IS_ORG_ANONYMOUS |
| 1 Sarkz Inc. | 567890 941 | Apple Dr. | 9482339123 | Willis Zackerman | Y |
| 2 Monsters Inc. | 987654 942 | Apple Dr. | 8472814999 | William Jockz | N |
| 3 Lightyears LLC. | 993123 8470 | Skynet Dr. | 3821042235 | Haley Redder | N |
| 4 Zinc Co. | 1942 8471 | Skynet Dr. | 9482221442 | Sarah Keller | Y |
| 5 Blizzard Entertainment | 339901 8472 | Skynet Dr. | 3528184421 | Josh Yanzak | N |
| 6 Jills Soaps | 564738 8472 | Skynet Dr. | 9283224233 | Bob Barker | Y |
| 7 Jordans Popcorn | 772056 8377 | Skynet Dr. | 9283885231 | Carol Geyzerz | N |
| 8 Michaels Kicks | 887201 9968 | Tech Lane | 3821242352 | Chris Brown | Y |
| 9 Michael Kors | 768301 844 | Gucci Lane | 9472385521 | Kim Laurel | N |
| 10 Nike Inc. | 113673 123 | Swoosh Rd. | 8572837712 | David McDavis | N |

6.10. Script file showing the testing of query 10



```
IP_JAVA_BLACK_EMILY [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Nov 13, 2017)
(18) Import: Enter new records from a data file until the file is empty (etc)
(19) Export: Retrieve names and mailing addresses of all people on the mail
(20) Quit.

Please enter the number to the corresponding task you want to run.
10
Please enter your input in the following format: SSN
123456

RESULTS:

Name of the client's doctor: Dr. Parker
Number of the client's doctor: 492395777

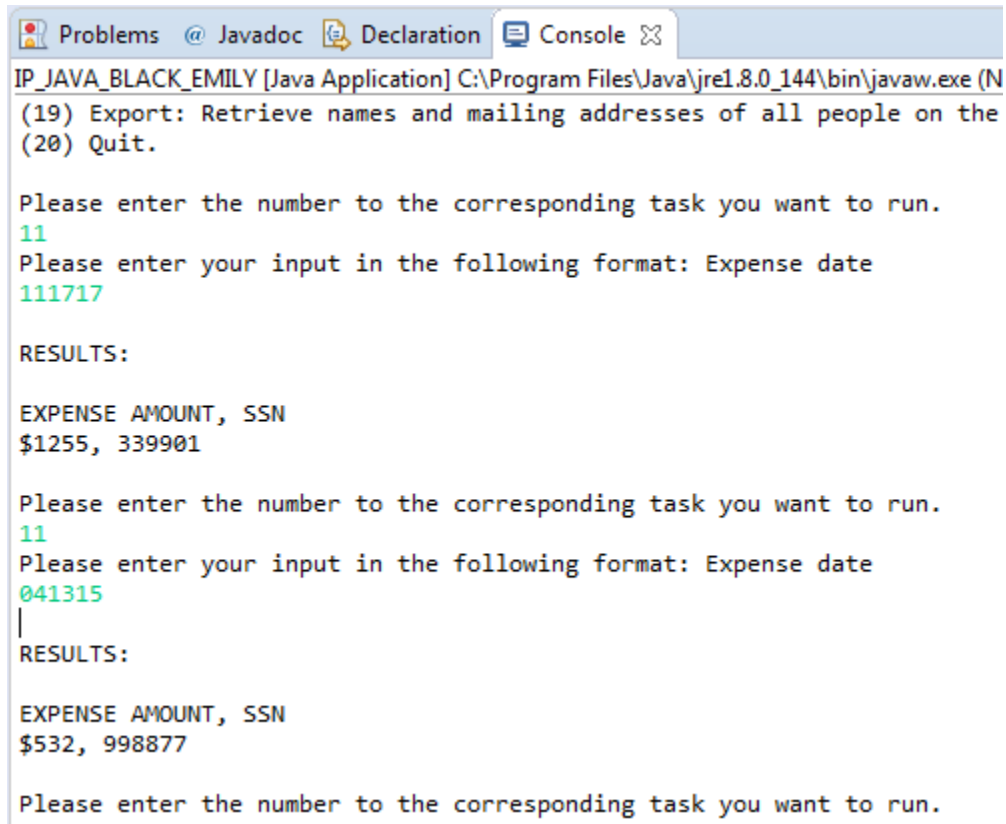
Please enter the number to the corresponding task you want to run.
10
Please enter your input in the following format: SSN
744210

RESULTS:

Name of the client's doctor: Dr. Jones
Number of the client's doctor: 2432432117

Please enter the number to the corresponding task you want to run.
```

6.11. Script file showing the testing of query 11



```
IP_JAVA_BLACK_EMILY [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (N
(19) Export: Retrieve names and mailing addresses of all people on the
(20) Quit.

Please enter the number to the corresponding task you want to run.
11
Please enter your input in the following format: Expense date
111717

RESULTS:

EXPENSE AMOUNT, SSN
$1255, 339901

Please enter the number to the corresponding task you want to run.
11
Please enter your input in the following format: Expense date
041315
|
RESULTS:

EXPENSE AMOUNT, SSN
$532, 998877

Please enter the number to the corresponding task you want to run.
```

6.12. Script file showing the testing of query 12

```

IP_JAVA_BLACK_EMILY [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Nov 13, 2017
(18) Import: Enter new teams from a data file until the file is empty (the user
(19) Export: Retrieve names and mailing addresses of all people on the mailing
(20) Quit.

Please enter the number to the corresponding task you want to run.
12
Please enter your input in the following format: SSN
567890

RESULTS:

SSN, TEAM NAME
112233, Team Yellow

Please enter the number to the corresponding task you want to run.
12
Please enter your input in the following format: SSN
123456

RESULTS:

SSN, TEAM NAME
884721, Team Gold

Please enter the number to the corresponding task you want to run.
|

```

6.13. Script file showing the testing of query 13

```

IP_JAVA_BLACK_EMILY [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Nov 13, 2017, 8:17:51 PM)
(15) For each team, retrieve the name and associated contact information of the volunteer that has
(16) Increase the salary by 10% of all employees to whom more than one team must report.
(17) Delete all clients who do not have health insurance and whose value of importance for transpo
(18) Import: Enter new teams from a data file until the file is empty (the user will be asked to e
(19) Export: Retrieve names and mailing addresses of all people on the mailing list and output the
(20) Quit.

Please enter the number to the corresponding task you want to run.
13

RESULTS:

NAME, EMAIL ADDRESS, MAILING ADDRESS, HOME PHONE, WORK PHONE, CELL PHONE
Sarah Johnson, sarah-johnson571@msu.edu, 492 Palace Place, 492319332, 5828329522, 5842392333

Please enter the number to the corresponding task you want to run.

```

6.14. Script file showing the testing of query 14

```

Problems @ Javadoc Declaration Console
IP_JAVA_BLACK_EMILY [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (No
(16) Increase the salary by 10% of all employees to whom more than one t
(17) Delete all clients who do not have health insurance and whose value
(18) Import: Enter new teams from a data file until the file is empty (t
(19) Export: Retrieve names and mailing addresses of all people on the n
(20) Quit.

```

Please enter the number to the corresponding task you want to run.

14

RESULTS:

Total Amount of Donations: \$28303

Contributors: Billy Willyard, Jimmy Barker, Lennard Kennard, Josh Kipps,

Is Donor Anonymous (True or False): Y, N, N, N,

Please enter the number to the corresponding task you want to run.

| | | | |
|---------|--------|------------------|-------------|
| CS4513 | | DONOR | |
| Columns | Data | Model | Constraints |
| | | Grants | Statistics |
| | | Sort.. | Filter: |
| | SSN | ISDONORANONYMOUS | |
| 1 | 238901 | Y | |
| 2 | 887201 | Y | |
| 3 | 772056 | N | |
| 4 | 768301 | Y | |
| 5 | 113673 | N | |
| 6 | 645221 | Y | |
| 7 | 339901 | N | |
| 8 | 564738 | N | |
| 9 | 163296 | N | |

CS4513 DONATION

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL

</

6.15. Script file showing the testing of query 15

```

Problems @ Javadoc Declaration Console
IP_JAVA_BLACK_EMILY [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Nov 13, 2017, 8:36:30 PM)
(17) Delete all clients who do not have health insurance and whose value of importance for transportation is
(18) Import: Enter new teams from a data file until the file is empty (the user will be asked to enter the i
(19) Export: Retrieve names and mailing addresses of all people on the mailing list and output them to a dat
(20) Quit.

Please enter the number to the corresponding task you want to run.
15
|
RESULTS:

EMAIL ADDRESS, MAILING ADDRESS, HOME PHONE, WORK PHONE, CELL PHONE, NAME, TEAM, HOURS
jack@gmail.com, 764 Crest Dr., 593021132, 492310492, 583291323, Jack Brown, Team Brown, 89
iysha@yahoo.com, 7482 Bucher Lane, 4029310423, 5029382311, 4029321132, Iysha Zimmerz, Team Gold, 87
jeffery.g@yahoo.com, 592 Bumpy Lane, 4928310422, 9582031140, 4827391042, Jeffery Gonzalez, Team White, 1
stacy_mockenz@outlook.com, 482 Horse Dr., 3928193922, 3819382291, 9482012234, Stacy Mockenz, Team Black, 0
stacy_mockenz@outlook.com, 482 Horse Dr., 3928193922, 3819382291, 9482012234, Stacy Mockenz, Team Yellow, 0
thedominator@wow.net, 0192 Oak Ridge Rd,, 3842831123, 8472931123, 5829381123, Don Black, Team Black, 0
thedominator@wow.net, 0192 Oak Ridge Rd,, 3842831123, 8472931123, 5829381123, Don Black, Team Yellow, 0

Please enter the number to the corresponding task you want to run.

```

6.16. Script file showing the testing of query 16

| Columns | | Data | Model | Constraints | Grants | Statistics | Triggers | Flash |
|---------|--------|--------|----------------|-------------|--------|------------|----------|-------|
| | | Sort.. | Filter: | | | | | |
| | SSN | SALARY | MARTIAL_STATUS | HIRE_DATE | | | | |
| 1 | 998877 | 56990 | Single | 91899 | | | | |
| 2 | 645221 | 95000 | Divorced | 92896 | | | | |
| 3 | 339901 | 43525 | Married | 91472 | | | | |
| 4 | 564738 | 102000 | Widow | 111389 | | | | |
| 5 | 163296 | 32000 | Single | 81109 | | | | |

Problems @ Javadoc Declaration Console

IP_JAVA_BLACK_EMILY [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (I

(16) Increase the salary by 10% of all employees to whom more than one
 (17) Delete all clients who do not have health insurance and whose va
 (18) Import: Enter new teams from a data file until the file is empty
 (19) Export: Retrieve names and mailing addresses of all people on the
 (20) Quit.

Please enter the number to the corresponding task you want to run.

16

Salaries successfully updated!

Please enter the number to the corresponding task you want to run.

CS4513 x EMPLOYEE x

Columns Data Model Constraints Grants Statistics Triggers Flash

Sort.. Filter:

| | SSN | SALARY | MARTIAL_STATUS | HIRE_DATE |
|---|--------|--------|----------------|-----------|
| 1 | 998877 | 62689 | Single | 91899 |
| 2 | 645221 | 104500 | Divorced | 92896 |
| 3 | 339901 | 43525 | Married | 91472 |
| 4 | 564738 | 112200 | Widow | 111389 |
| 5 | 163296 | 35200 | Single | 81109 |

6.17. Script file showing the testing of query 17

CS4513 x INSURANCE_POLICY x

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes

Sort.. Filter:

| | POLICY_ID | SSN | PROVIDER_ID | PROVIDER_ADDRESS | PROVIDER_TYPE |
|---|-----------|--------|-------------|------------------|---------------|
| 1 | 948283 | 744210 | 5433 123 | Health Lane | Health |
| 2 | 623423 | 987654 | 6534 456 | Doctor Dr. | Health |
| 3 | 643634 | 321098 | 2342 789 | Insurance Rd. | Health |
| 4 | 423623 | 567890 | (null) 472 | Hillcrest Peak | Life |
| 5 | 652345 | 123456 | (null) 4826 | Chestnut Rd. | Auto |

CS4513 x NEED x

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL

Sort.. Filter:

| | SSN | NEED | VALUE_O... |
|----|--------|----------------|------------|
| 1 | 744210 | Transportation | 9 |
| 2 | 567890 | Food | 3 |
| 3 | 567890 | Shopping | 4 |
| 4 | 744210 | Shopping | 7 |
| 5 | 321098 | Housekeeping | 2 |
| 6 | 987654 | Shopping | 9 |
| 7 | 123456 | Visiting | 1 |
| 8 | 321098 | Visiting | 3 |
| 9 | 567890 | Transportation | 5 |
| 10 | 123456 | Transportation | 1 |
| 11 | 987654 | Yard work | 8 |
| 12 | 123456 | Food | 5 |

CS4513 x CLIENT x

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL

Sort.. Filter:

| | SSN | DOCTORS_NAME | ATTORNEYS_NAME | DOCTORS_NUMBER | ATTORNEYS_NUMBER | DATE_OF_ASSIGNMENT |
|---|--------|--------------|----------------|----------------|------------------|--------------------|
| 1 | 123456 | Dr. Parker | Jack Johnson | 492395777 | 492831233 | 90714 |
| 2 | 744210 | Dr. Jones | Sadie Brown | 2432432117 | 3918418499 | 91501 |
| 3 | 567890 | Dr. Jones | Sadie Brown | 2432432117 | 3918418499 | 91501 |
| 4 | 987654 | Dr. Jones | Sadie Brown | 2432432117 | 3918418499 | 91501 |
| 5 | 321098 | Dr. Jones | Sadie Brown | 2432432117 | 3918418499 | 91501 |

Problems @ Javadoc Declaration Console

IP_JAVA_BLACK_EMILY [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Nov 1

(16) Increase the salary by 10% of all employees to whom more than one te
 (17) Delete all clients who do not have health insurance and whose value
 (18) Import: Enter new teams from a data file until the file is empty (th
 (19) Export: Retrieve names and mailing addresses of all people on the ma
 (20) Quit.

Please enter the number to the corresponding task you want to run.

17

Clients successfully updated!

Please enter the number to the corresponding task you want to run.

| CS4513 x CLIENT x | | Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL | | | | |
|-------------------|--------|---|----------------|----------------|------------------|--------------------|
| | | Sort.. Filter: | | | | |
| | SSN | DOCTORS_NAME | ATTORNEYS_NAME | DOCTORS_NUMBER | ATTORNEYS_NUMBER | DATE_OF_ASSIGNMENT |
| 1 | 744210 | Dr. Jones | Sadie Brown | 2432432117 | 3918418499 | 91501 |
| 2 | 567890 | Dr. Jones | Sadie Brown | 2432432117 | 3918418499 | 91501 |
| 3 | 987654 | Dr. Jones | Sadie Brown | 2432432117 | 3918418499 | 91501 |
| 4 | 321098 | Dr. Jones | Sadie Brown | 2432432117 | 3918418499 | 91501 |

6.18. Script file showing the testing of the import and export options

Import:

The screenshot illustrates the process of importing data from an Excel file into a database. It is divided into three main sections:

Excel Spreadsheet Data

| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|-------------|---------|--------|----------------|---|---|---|---|---|---|---|---|
| 1 | Team Green | Finance | 394210 | Mitchell Jones | | | | | | | | |
| 2 | Team Orange | Stocks | 392312 | Barry Barker | | | | | | | | |

File Explorer Window

The file explorer shows the 'Workspace' directory containing a folder named 'IndividualProject'. Inside this folder, the file 'myInput' (Microsoft Excel C...) is selected.

SQL Table Data

The bottom section shows a SQL table named 'TEAM' with the following columns: TEAM_NAME, TEAM_TYPE, DATE_OF_FORMATION, and TEAM_LEAD. The table contains five rows of data:

| | TEAM_NAME | TEAM_TYPE | DATE_OF_FORMATION | TEAM_LEAD |
|---|-------------|------------|-------------------|-----------------|
| 1 | Team Black | Operations | 72709 | Jerry Brown |
| 2 | Team Brown | Networking | 80114 | Donald Jones |
| 3 | Team Yellow | Marketing | 31307 | Barbara Jenkins |
| 4 | Team White | Finance | 50214 | Barry Jenkins |
| 5 | Team Gold | Resources | 72413 | Bob Jones |

Problems Javadoc Declaration Console

IP JAVA BLACK EMILY [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Nov 13, 2017, 9:22:16 PM)

(9) Enter a new organization and associate it with several donations.
 (10) Retrieve the name and phone number of the doctor of a particular client.
 (11) Retrieve the total amount of expenses charged by each employee for a particular period of time. The list should be sorted by the total am
 (12) Retrieve the list of volunteers that are members of teams that support a particular client.
 (13) Retrieve the names and contact information of the clients that are supported by teams sponsored by an organization whose name starts with
 (14) Retrieve the name and total amount donated by donors that are also employees. The list should be sorted by the total amount of the donat
 (15) For each team, retrieve the name and associated contact information of the volunteer that has worked the most total hours between March a
 (16) Increase the salary by 10% of all employees to whom more than one team must report.
 (17) Delete all clients who do not have health insurance and whose value of importance for transportation is less than 5.
 (18) Import: Enter new teams from a data file until the file is empty (the user will be asked to enter the input file name).
 (19) 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
 (20) Quit.

Please enter the number to the corresponding task you want to run.

18

Please enter the name of your CSV data file to import (Ex: myInputFile.csv). Please make sure your file is located in the project directory.

myInput.csv

myInput.csv






Total records inserted in bulk from CSV file: 2

New teams successfully imported!

Please enter the number to the corresponding task you want to run.

CS4513 TEAM

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Det

     | Sort.. Filter:

| | TEAM_NAME | TEAM_TYPE | DATE_OF_FORMATION | TEAM_LEAD | |
|---|-------------|------------|-------------------|-----------------|--|
| 1 | Team Black | Operations | 72709 | Jerry Brown | |
| 2 | Team Brown | Networking | 80114 | Donald Jones | |
| 3 | Team Yellow | Marketing | 31307 | Barbara Jenkins | |
| 4 | Team White | Finance | 50214 | Barry Jenkins | |
| 5 | Team Gold | Resources | 72413 | Bob Jones | |
| 6 | Team Green | Finance | 394210 | Mitchell Jones | |
| 7 | Team Orange | Stocks | 392312 | Barry Barker | |

Export:

| CS4513 PERSON | | | | | | | | | |
|---|--------|-------------------|---------------|-------|--------|--------|-------------|-----------------|----------|
| Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL | | | | | | | | | |
| Sort: Filter: | | | | | | | | | |
| | SSN | EMAIL_ADDRESS | NAME | DOB | RACE | GENDER | PROFESSION | ISONMAILINGL... | ISAFFILI |
| 1 | 744210 | zack@ou.edu | Zack Morenio | 91893 | White | M | EE | Y | N |
| 2 | 768301 | hillard@hot... | Hillard Keya | 12797 | Afr... | M | CS/EE | Y | Y |
| 3 | 238901 | yanzak@hotmail... | Jill Yanzak | 52399 | White | F | EE | Y | N |
| 4 | 339901 | jimmy-b@yahoo... | Jimmy Barker | 11... | White | M | ME | Y | Y |
| 5 | 645221 | joshk@yahoo... | Josh Kipps | 51857 | White | M | EE | Y | N |
| 6 | 993123 | jeffery.g@y... | Jeffery Go... | 71797 | His... | M | Literature | Y | Y |
| 7 | 665743 | jack@gmail.com | Jack Brown | 90592 | White | M | CE | Y | N |
| 8 | 567890 | bill@gmail.edu | Bill Gardener | 10476 | Afr... | M | Physics | Y | Y |
| 9 | 1942 | stacy_mock... | Stacy Mockenz | 51657 | White | F | English | N | Y |
| 10 | 112233 | thedominato... | Don Black | 84... | Afr... | M | Cook | N | N |
| 11 | 998877 | yyija@yahoo... | Yyija Mathias | 40383 | Afr... | F | CS | N | N |
| 12 | 772056 | 847291@ou.edu | Buddy Esther | 31299 | Afr... | F | Biomedical | N | Y |
| 13 | 163296 | billyw@yaho... | Billy Will... | 71492 | White | M | CS/CE | N | N |
| 14 | 123456 | emily@ou.edu | Emily Black | 12... | White | F | CS | N | N |
| 15 | 564738 | lennard423@... | Lennard Ke... | 40567 | Asian | M | ME/CE | N | Y |
| 16 | 321098 | sarah-johns... | Sarah Johnson | 62101 | Mex... | F | Human Re... | N | N |
| 17 | 987654 | jerry.m.jon... | Jerry Mich... | 62101 | Asian | M | Marketing | N | Y |
| 18 | 887201 | sarah-h31@h... | Sarah Haah | 91295 | Mex... | F | CCE | N | Y |
| 19 | 113673 | 133tkid090@... | Mathew Joller | 12... | White | M | Business | N | Y |
| 20 | 884721 | iysha@yahoo... | Iysha Zimmerz | 71487 | Indian | F | Art | N | N |

The screenshot shows an IDE window with a file explorer overlay. The file explorer is displaying the contents of a workspace named 'IndividualProject'. The file list includes:

| Name | Date modified | Type | Size |
|------------|----------------------|----------------------|------|
| .settings | 10/8/2017 7:29 PM | File folder | |
| bin | 11/12/2017 11:51 ... | File folder | |
| src | 11/12/2017 12:29 ... | File folder | |
| .classpath | 11/12/2017 12:02 ... | CLASSPATH File | 1 KB |
| .project | 11/4/2017 2:43 PM | PROJECT File | 1 KB |
| myInput | 11/13/2017 9:25 PM | Microsoft Excel C... | 1 KB |

Below the file explorer, a snippet of Java code is visible:

```

718 e.printStackTrace();
719 }
720 stmt.registerOutParameter(1, OracleTypes.CURSOR);
721 stmt.execute();
722 ResultSet rs = (ResultSet) stmt.getObject(1);
723 // Writes all data from ResultSet to output file. Includes column names.
724

```

The bottom of the IDE shows the 'Console' tab with the following text:

IP_JAVA_BLACK_EMILY [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Nov 13, 2017, 9:38:28 PM)

- (8) Enter a new donor and associate him or her with several donations.
- (9) Enter a new organization and associate it with several donations.
- (10) Retrieve the name and phone number of the doctor of a particular client.
- (11) Retrieve the total amount of expenses charged by each employee for a particular period of time. The list should be sorted
- (12) Retrieve the list of volunteers that are members of teams that support a particular client.
- (13) Retrieve the names and contact information of the clients that are supported by teams sponsored by an organization whose n
- (14) Retrieve the name and total amount donated by donors that are also employees. The list should be sorted by the total amou
- (15) For each team, retrieve the name and associated contact information of the volunteer that has worked the most total hours
- (16) Increase the salary by 10% of all employees to whom more than one team must report.
- (17) Delete all clients who do not have health insurance and whose value of importance for transportation is less than 5.
- (18) Import: Enter new teams from a data file until the file is empty (the user will be asked to enter the input file name).
- (19) Export: Retrieve names and mailing addresses of all people on the mailing list and output them to a data file instead of s
- (20) Quit.

Please enter the number to the corresponding task you want to run.

The screenshot shows an IDE with a file explorer window open over a project named 'IndividualProject'. The file explorer displays the following files and folders:

| Name | Date modified | Type | Size |
|------------|----------------------|----------------------|------|
| .settings | 10/8/2017 7:29 PM | File folder | |
| bin | 11/12/2017 11:51 ... | File folder | |
| src | 11/12/2017 12:29 ... | File folder | |
| .classpath | 11/12/2017 12:02 ... | CLASSPATH File | 1 KB |
| .project | 11/4/2017 2:43 PM | PROJECT File | 1 KB |
| myExports | 11/13/2017 9:38 PM | Microsoft Excel C... | 1 KB |
| myInput | 11/13/2017 9:25 PM | Microsoft Excel C... | 1 KB |

Below the file explorer, a snippet of Java code is visible:

```

718         e.printStackTrace();
719     }
720     stmt.registerOutParameter(1, OracleTypes.CURSOR);
721     stmt.execute();
722     ResultSet rs = (ResultSet) stmt.getObject(1);
723     // Writes all data from ResultSet to output file. Includes column names.
724     ...

```

The console window at the bottom shows the following output:

```

IP_JAVA_BLACK_EMILY [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Nov 13, 2017, 9:38:28 PM)
(15) For each team, retrieve the name and associated contact information of the volunteer that has worked the most total hours betw
(16) Increase the salary by 10% of all employees to whom more than one team must report.
(17) Delete all clients who do not have health insurance and whose value of importance for transportation is less than 5.
(18) Import: Enter new teams from a data file until the file is empty (the user will be asked to enter the input file name).
(19) Export: Retrieve names and mailing addresses of all people on the mailing list and output them to a data file instead of screen
(20) Quit.

Please enter the number to the corresponding task you want to run.
19
Please enter the name of your CSV data file to export (Ex: myExportFile.csv). The file will be placed in the project directory.
myExports.csv

Data successfully exported to myExports.csv

Please enter the number to the corresponding task you want to run.

```

| | A | B | C | D | E | F | G | H | I |
|----|------------------|-----------------------|---|---|---|---|---|---|---|
| 1 | NAME | MAILING_ADDRESS | | | | | | | |
| 2 | Zack Morenio | 8431 West Drive | | | | | | | |
| 3 | Bill Gardener | Three Partners Place | | | | | | | |
| 4 | Jack Brown | 764 Crest Dr. | | | | | | | |
| 5 | Jeffery Gonzalez | 592 Bumpy Lane | | | | | | | |
| 6 | Josh Kipps | 95834 Blue Creek Lane | | | | | | | |
| 7 | Jimmy Barker | 623 Crystal Cr. | | | | | | | |
| 8 | Jill Yanzak | 726 Riverwind Dr. | | | | | | | |
| 9 | Hillard Keya | 113 Complex Road | | | | | | | |
| 10 | | | | | | | | | |
| 11 | | | | | | | | | |
| 12 | | | | | | | | | |
| 13 | | | | | | | | | |
| 14 | | | | | | | | | |
| 15 | | | | | | | | | |
| 16 | | | | | | | | | |
| 17 | | | | | | | | | |
| 18 | | | | | | | | | |

6.19. Script file showing the testing of three types of errors

Unique constraint violated, PK already exists:

```

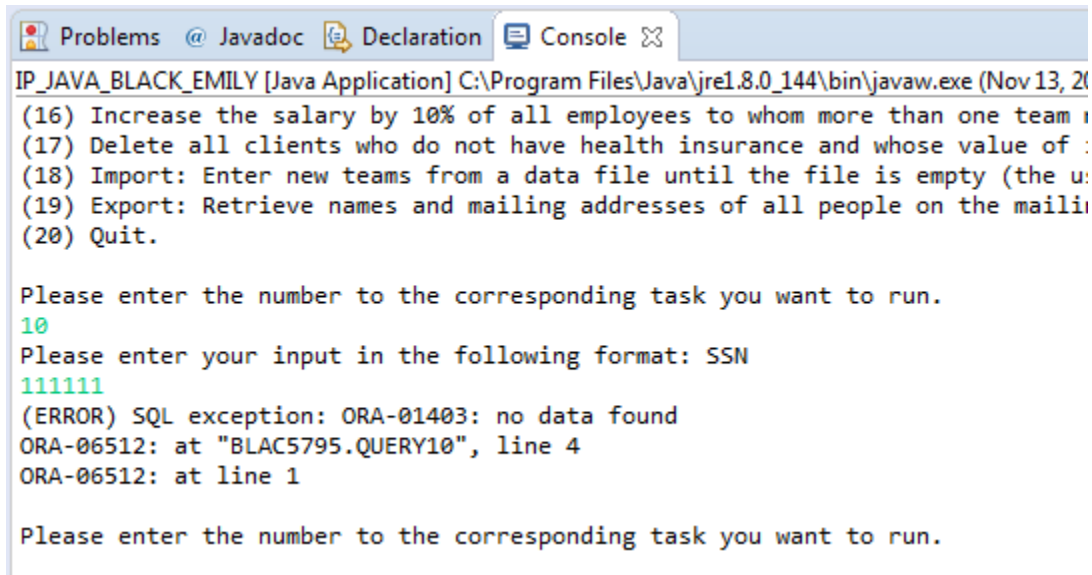
Problems @ Javadoc Declaration Console
IP_JAVA_BLACK_EMILY [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Nov 13, 2017, 9:51:28 PM)
(16) Increase the salary by 10% of all employees to whom more than one team must report
(17) Delete all clients who do not have health insurance and whose value of importance is less than 10
(18) Import: Enter new teams from a data file until the file is empty (the user will be prompted to stop)
(19) Export: Retrieve names and mailing addresses of all people on the mailing list and save to a file
(20) Quit.

Please enter the number to the corresponding task you want to run.
1
Please enter your input in the following format: Team name, Team type, Date of format:
Team Gold, Marketing, 031416, Sarah Daniels
(ERROR) SQL exception: ORA-00001: unique constraint (BLAC5795.SYS_C00181764) violated
ORA-06512: at "BLAC5795.QUERY1", line 4
ORA-06512: at line 1

Please enter the number to the corresponding task you want to run.

```

No data found error:



```

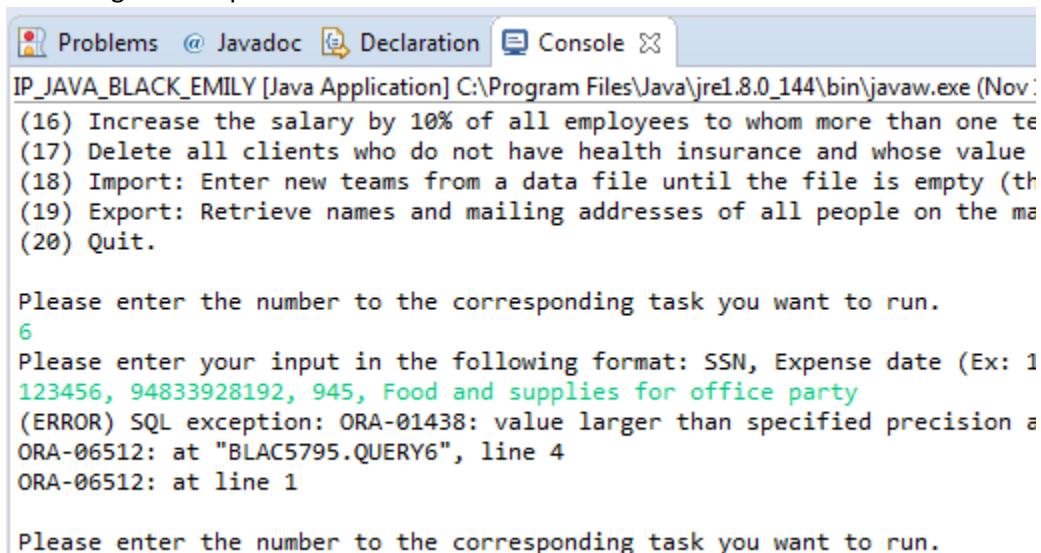
Problems @ Javadoc Declaration Console
IP_JAVA_BLACK_EMILY [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Nov 13, 2016)
(16) Increase the salary by 10% of all employees to whom more than one team
(17) Delete all clients who do not have health insurance and whose value of :
(18) Import: Enter new teams from a data file until the file is empty (the u
(19) Export: Retrieve names and mailing addresses of all people on the maili
(20) Quit.

Please enter the number to the corresponding task you want to run.
10
Please enter your input in the following format: SSN
111111
(ERROR) SQL exception: ORA-01403: no data found
ORA-06512: at "BLAC5795.QUERY10", line 4
ORA-06512: at line 1

Please enter the number to the corresponding task you want to run.

```

Value larger than specified attribute size:



```

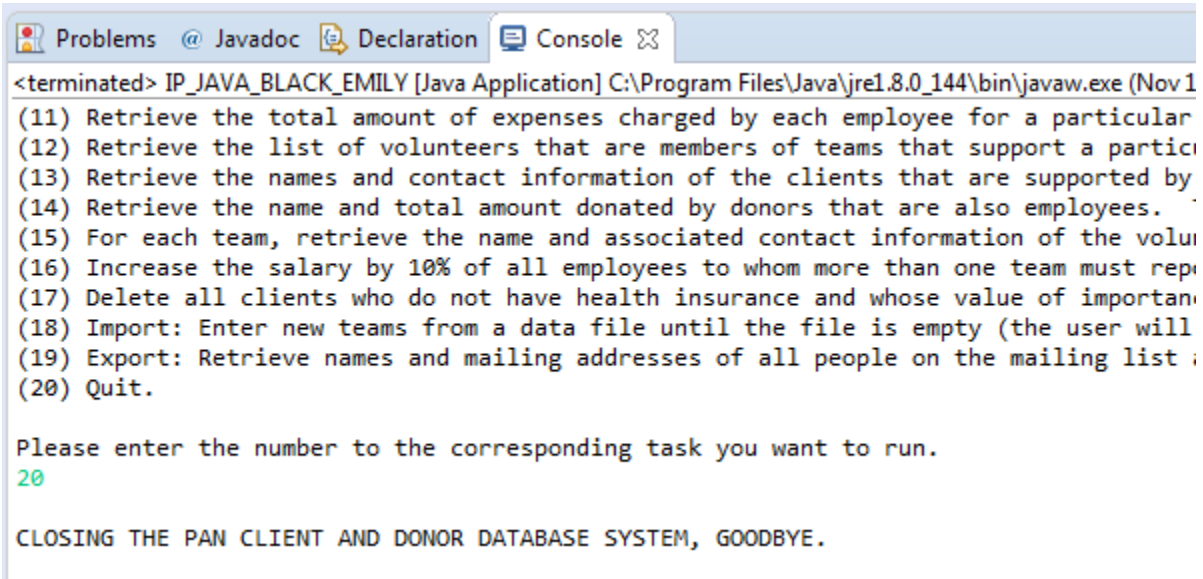
Problems @ Javadoc Declaration Console
IP_JAVA_BLACK_EMILY [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Nov 13, 2016)
(16) Increase the salary by 10% of all employees to whom more than one te
(17) Delete all clients who do not have health insurance and whose value
(18) Import: Enter new teams from a data file until the file is empty (th
(19) Export: Retrieve names and mailing addresses of all people on the ma
(20) Quit.

Please enter the number to the corresponding task you want to run.
6
Please enter your input in the following format: SSN, Expense date (Ex: 1
123456, 94833928192, 945, Food and supplies for office party
(ERROR) SQL exception: ORA-01438: value larger than specified precision a
ORA-06512: at "BLAC5795.QUERY6", line 4
ORA-06512: at line 1

Please enter the number to the corresponding task you want to run.

```


6.20. Script file showing the testing of the quit option



```
Problems @ Javadoc Declaration Console X
<terminated> IP_JAVA_BLACK_EMILY [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Nov 1
(11) Retrieve the total amount of expenses charged by each employee for a particular
(12) Retrieve the list of volunteers that are members of teams that support a partici
(13) Retrieve the names and contact information of the clients that are supported by
(14) Retrieve the name and total amount donated by donors that are also employees.
(15) For each team, retrieve the name and associated contact information of the volun
(16) Increase the salary by 10% of all employees to whom more than one team must rep
(17) Delete all clients who do not have health insurance and whose value of importan
(18) Import: Enter new teams from a data file until the file is empty (the user will
(19) Export: Retrieve names and mailing addresses of all people on the mailing list
(20) Quit.

Please enter the number to the corresponding task you want to run.
20

CLOSING THE PAN CLIENT AND DONOR DATABASE SYSTEM, GOODBYE.
```