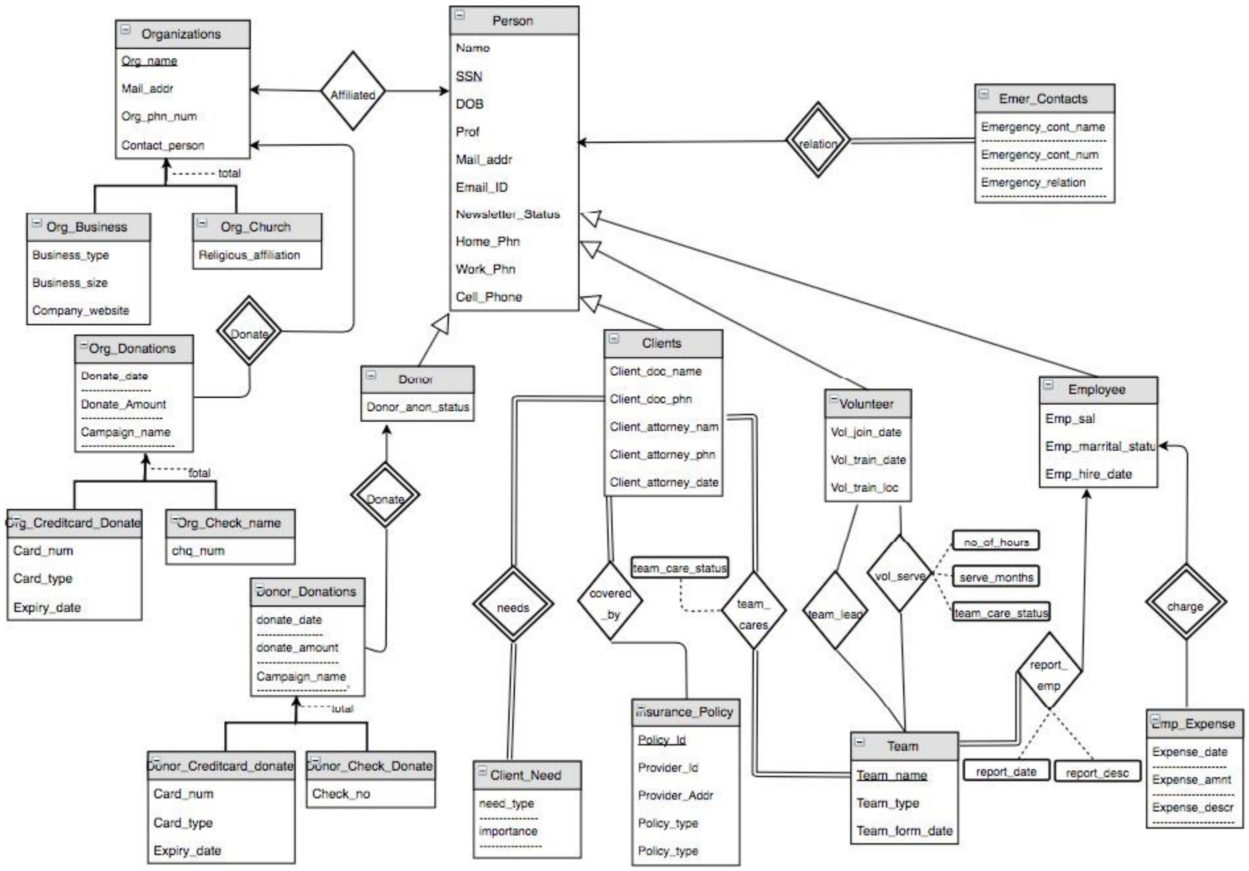
CLIENT AND DONOR DATABASE SYSTEM

Project

Author: SUMITH GANNARAPU

TASK 1: 1.1 ER DIAGRAM



TASK 1.2: RELATIONAL SCHEMA:

Relational Schema for tables/relations:

Tables:

Person(name,ssn,dob,race,gender,proff,mail_addr,email_id,newsletter_status,home_phn,work_phn,cell_phn)

Emergency_contact(ssn,emergency_cont_name,emergency_cont_num,emergency_relation)

Clients(client_ssn,client_doc_name,client_doc_phn,client_attorney_name,client_attorney_phn, client_assign_date)

Volunteer(vol_ssn,vol_join_date,vol_train_date,vol_train_loc)

Employee(emp_ssn,emp_sal,emp_marrital_status,emp_hire_date)

Donor(donor_ssn,donor_anon_status)

Insurance_policy(policy_id,provider_id,provider_addr,policy_type)

Team(team_name,team_type,team_form_date)

Emp_expense(emp_ssn,expense_date,expense_amnt,expense_descr)

Organizations(org_name,mail_addr,org_phn_num,contact_person,org_anon_status)

Organization_business(org_name,business_type,business_size,company_website)

Org_church(org_name,religious_affiliation)

donor_donations(donor_ssn,donate_date,donate_amount,campaign_name)

Donor_creditcard_donate(donor_ssn,donate_date,donate_amount,campaign_name,card_num,card_type,expiry_date)

Donor_check_donate(donor_ssn,donate_date,donate_amount,campaign_name,check_num)

Org_donations(org_name,donate_date,donate_amount,campaign_name)

Org_creditcard_donate(org_name,donate_date,donate_amount,campaign_name,card_num,card_type,expiry_date)

Org_check_donate(org_name,donate_date,donate_amount,campaign_name,chq_number)

Relations:

Client_need(client_ssn,need_type,importance)

Covered_by(client_ssn,policy_id)

Team_cares(client_ssn,team_name,team_care_status)

Report_emp(emp_ssn,team_name,report_date,report_desc)

Vol_serve(vol_ssn,team_name,no_of_hours,serve_months,team_care_status)

Team_lead(vol_ssn,team_name)

Affiliated(ssn,org_name)

Sponsors(team_name,org_name)

Task 2.1:

Indexing-Storage Structure:

Table Name	Types of Queries	Search Key(s)	Frequency	File Organization
Person	13. Random Search 14. Random Search 15. Random Search 16. Random Search	ssn ssn ssn name	1/week 1/week 1/month 1/year	Dynamic Hashing
Clients	2.Insert 10.Random Search 13.Range Search 17.Deletion	 Client _ssn Client_ssn Client_ssn	1/week 1/week 1/week 4/year	Dynamic Hashing
Employee	5. Insertion 6.Updation 14.Random Search 11.Range Search 16. Updation	 Emp_ssn Emp_ssn Emp_ssn	1/day 1/day 1/week 1/month 1/year	Dynamic Extendable Hashing
Volunteer	3.Insertion 4.Insertion 12.Random Search 15.Range Search	 Vol_ssn Vol_ssn	2/month 30/month 4/year 1/month	Index-sequential.
Donor	8.Insertion 14.Random Search	 Donor_ssn	1/day 1/week	Dynamic extendable hashing
Organization	7.Insertion 9.Insertion 13.Range Search	 Org_Name	2/week 1/day 1/week	Sequential storage
Report_emp	16.Updation	E_ssn	1/year	Heap.

Team	1.Insertion		1/month	B+Tree
	2.Insertion		1/week	
	3.Insertion		2/month	
	4.Insertion		30/month	
	5.Insertion		1/year	
	7.Insertion		2/week	
	12.Random Search	Team_name	4/year	
	13.Range Search	Team_name	1/week	
	15.Range Search	Team_name	1/month	
Vol_serve	3.Insertion		2/month	Indexed Sequential search
	4.Updation	No_of_hours	30/month	
	12.Random Search	Client SSN	4/year	
	15.Range Search	Serve_month	1/month	
Sponsors	7.Insertion		2/week	Sequential
	13.Range Search	Org Name	4/year	
Donor Donations	8.Insertion		1/day	Heap
Org Donation	9. Insertion		1/day	Heap
Covered_by	10.Random Search	policy_id	1/week	Hash File Structure
	17.Deletion	Client_ssn	4/year	
Emp_expenses	11.Range Search	Emp_expense	1/month	Indexed sequential search.
	6.Updation		1/day	
Insurance_policy	17. Deletion	Policy_id	4/year	Sequential file storage
Team_cares	2. Insertion		1/week	Range search
	13. Range Search	Team_Name, Client_ssn	4/year	

TASK 2.2.Discussion of storage structures for tables (Oracle 12c)

For all my tables, I used the default storage structure implementations . Since Oracle automatically create primary indices on Primary key column of table. I will create few secondary indices columns for the tables to retrieve the records efficiently.

Whenever a table is created by the user, the best optimum storage structure is chosen and implemented by the Oracle 12 according to the table information and constraints on it.

Oracle does not support all different types of storage structures. It does support B-Tree Storage Structure and thus, for all the relational tables, the storage structure implemented on Oracle B-Tree

The ones I will create are as follows:

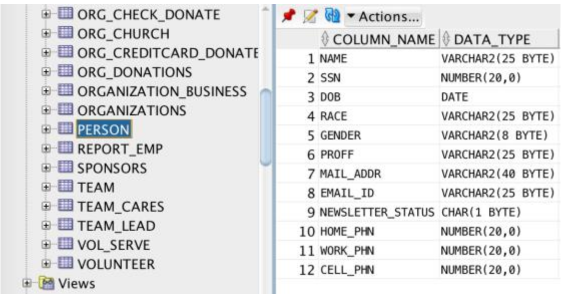
Table	Attribute	Index
Team	Team_type	B-Tree Index on Team_type
Person	Race	B-Tree Index on Race
Insurance_Policy	Policy_type	B-Tree Index on Policy_type
Organization_Business	Business_type	B-Tree Index on Business_type
Volunteer	Vol_train_loc	B-Tree Index on Vol_train_loc

Secondary Indices:

```
CREATE INDEX TEAM_INDEX ON TEAM(TEAM_TYPE);
CREATE INDEX PERSON_RACE_INDEX ON PERSON(RACE);
CREATE INDEX POLICY_TYPE_INDEX ON INSURANCE_POLICY(POLICY_TYPE);
CREATE INDEX BUSINESS_TYPE_INDEX ON ORGANIZATION_BUSINESS(BUSINESS_TYPE);
CREATE INDEX VOL_TRAIN_LOC_INDEX ON VOLUNTEER(VOL_TRAIN_LOC);
```

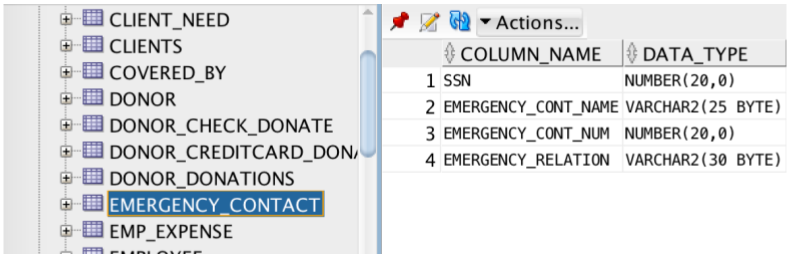
TASK 3:
<u>CREATING TABLES IN SQL</u>

```
/* CREATING TABLE FOR PERSON */
CREATE TABLE PERSON
(
NAME VARCHAR2(25),
SSN NUMBER(20,0) PRIMARY KEY,
DOB DATE,
RACE VARCHAR2(25),
GENDER VARCHAR2(8),
PROFF VARCHAR2(25),
MAIL_ADDR VARCHAR2(40),
EMAIL_ID VARCHAR2(25),
NEWSLETTER_STATUS CHAR(1),
HOME_PHN NUMBER(20,0),
WORK_PHN NUMBER(20,0),
CELL_PHN NUMBER(20,0)
);
```



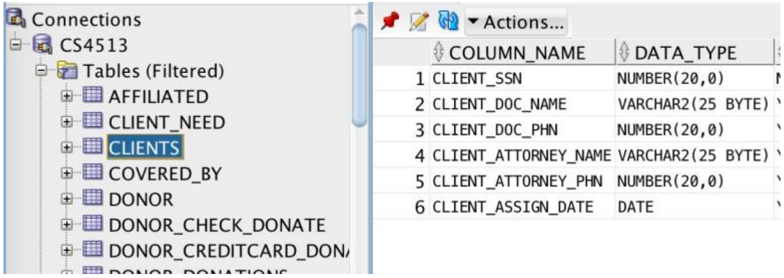
	COLUMN_NAME	DATA_TYPE
1	NAME	VARCHAR2(25 BYTE)
2	SSN	NUMBER(20,0)
3	DOB	DATE
4	RACE	VARCHAR2(25 BYTE)
5	GENDER	VARCHAR2(8 BYTE)
6	PROFF	VARCHAR2(25 BYTE)
7	MAIL_ADDR	VARCHAR2(40 BYTE)
8	EMAIL_ID	VARCHAR2(25 BYTE)
9	NEWSLETTER_STATUS	CHAR(1 BYTE)
10	HOME_PHN	NUMBER(20,0)
11	WORK_PHN	NUMBER(20,0)
12	CELL_PHN	NUMBER(20,0)

```
/* CREATING TABLE FOR EMERGENCY_CONTACT */
CREATE TABLE EMERGENCY_CONTACT
(
SSN NUMBER(20,0),
EMERGENCY_CONT_NAME VARCHAR2(25),
EMERGENCY_CONT_NUM NUMBER(20,0),
EMERGENCY_RELATION VARCHAR2(30),
PRIMARY KEY(EMERGENCY_CONT_NAME,EMERGENCY_CONT_NUM,EMERGENCY_RELATION,SSN),
FOREIGN KEY(SSN) REFERENCES PERSON ON DELETE CASCADE
);
```



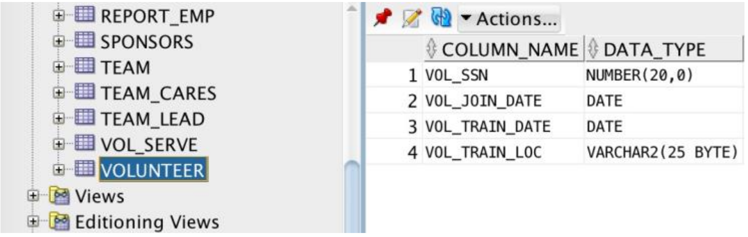
	COLUMN_NAME	DATA_TYPE
1	SSN	NUMBER(20,0)
2	EMERGENCY_CONT_NAME	VARCHAR2(25 BYTE)
3	EMERGENCY_CONT_NUM	NUMBER(20,0)
4	EMERGENCY_RELATION	VARCHAR2(30 BYTE)

```
/* CREATING TABLE FOR CLIENTS */
CREATE TABLE CLIENTS
(
CLIENT_SSN NUMBER(20,0) PRIMARY KEY,
CLIENT_DOC_NAME VARCHAR2(25),
CLIENT_DOC_PHN NUMBER(20,0),
CLIENT_ATTORNEY_NAME VARCHAR2(25),
CLIENT_ATTORNEY_PHN NUMBER(20,0),
CLIENT_ASSIGN_DATE DATE,
FOREIGN KEY(CLIENT_SSN) REFERENCES PERSON(SSN) ON DELETE CASCADE
);
```



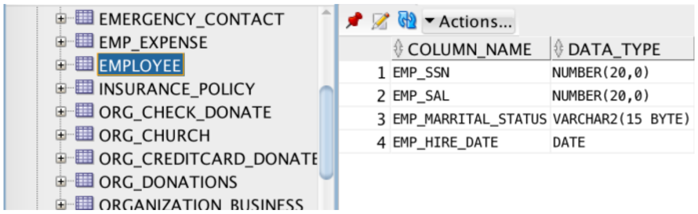
	COLUMN_NAME	DATA_TYPE
1	CLIENT_SSN	NUMBER(20,0)
2	CLIENT_DOC_NAME	VARCHAR2(25 BYTE)
3	CLIENT_DOC_PHN	NUMBER(20,0)
4	CLIENT_ATTORNEY_NAME	VARCHAR2(25 BYTE)
5	CLIENT_ATTORNEY_PHN	NUMBER(20,0)
6	CLIENT_ASSIGN_DATE	DATE

```
/* CREATING TABLE FOR VOLUNTEER */
CREATE TABLE VOLUNTEER
(
VOL_SSN NUMBER(20,0) PRIMARY KEY,
VOL_JOIN_DATE DATE,
VOL_TRAIN_DATE DATE,
VOL_TRAIN_LOC VARCHAR2(25),
FOREIGN KEY(VOL_SSN) REFERENCES PERSON(SSN) ON DELETE CASCADE
);
```



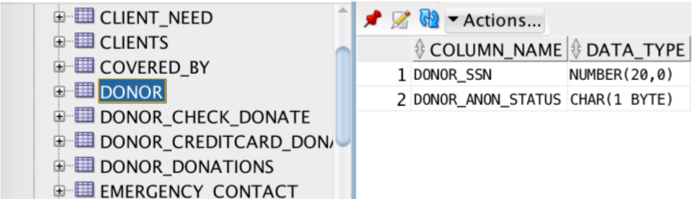
	COLUMN_NAME	DATA_TYPE
1	VOL_SSN	NUMBER(20,0)
2	VOL_JOIN_DATE	DATE
3	VOL_TRAIN_DATE	DATE
4	VOL_TRAIN_LOC	VARCHAR2(25 BYTE)

```
/* CREATING TABLE FOR EMPLOYEE */
CREATE TABLE EMPLOYEE
(
EMP_SSN NUMBER(20,0) PRIMARY KEY,
EMP_SAL NUMBER(20,0),
EMP_MARRITAL_STATUS VARCHAR2(15),
EMP_HIRE_DATE DATE,
FOREIGN KEY(EMP_SSN) REFERENCES PERSON(SSN) ON DELETE CASCADE
);
```



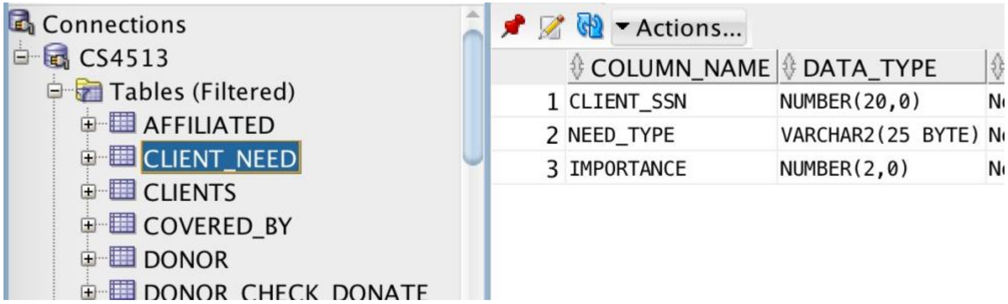
	COLUMN_NAME	DATA_TYPE
1	EMP_SSN	NUMBER(20,0)
2	EMP_SAL	NUMBER(20,0)
3	EMP_MARRITAL_STATUS	VARCHAR2(15 BYTE)
4	EMP_HIRE_DATE	DATE

```
/* CREATING TABLE FOR DONOR */
CREATE TABLE DONOR
(
DONOR_SSN NUMBER(20,0) PRIMARY KEY,
DONOR_ANON_STATUS CHAR(1),
FOREIGN KEY(DONOR_SSN) REFERENCES PERSON(SSN) ON DELETE CASCADE
);
```



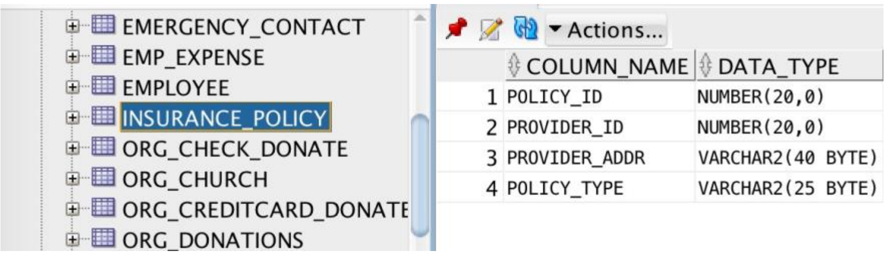
	COLUMN_NAME	DATA_TYPE
1	DONOR_SSN	NUMBER(20,0)
2	DONOR_ANON_STATUS	CHAR(1 BYTE)

```
/* CREATING TABLE FOR CLIENT_NEED */
CREATE TABLE CLIENT_NEED
(
CLIENT_SSN   NUMBER(20,0),
NEED_TYPE   VARCHAR2(25),
IMPORTANCE NUMBER(2),
PRIMARY KEY(CLIENT_SSN,NEED_TYPE,IMPORTANCE),
FOREIGN KEY(CLIENT_SSN) REFERENCES CLIENTS ON DELETE CASCADE
);
```



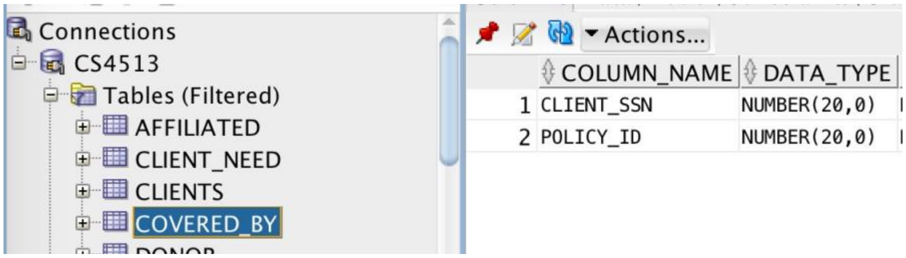
	COLUMN_NAME	DATA_TYPE	
1	CLIENT_SSN	NUMBER(20,0)	N
2	NEED_TYPE	VARCHAR2(25 BYTE)	N
3	IMPORTANCE	NUMBER(2,0)	N

```
/* CREATING TABLE FOR INSURANCE_POLICY */
CREATE TABLE INSURANCE_POLICY
(
POLICY_ID NUMBER(20,0) PRIMARY KEY,
PROVIDER_ID NUMBER(20,0),
PROVIDER_ADDR VARCHAR2(40),
POLICY_TYPE VARCHAR2(25)
);
```



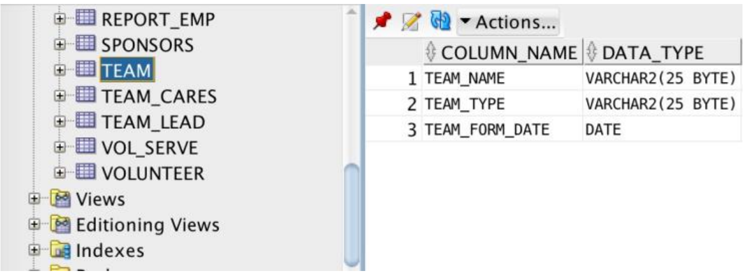
	COLUMN_NAME	DATA_TYPE
1	POLICY_ID	NUMBER(20,0)
2	PROVIDER_ID	NUMBER(20,0)
3	PROVIDER_ADDR	VARCHAR2(40 BYTE)
4	POLICY_TYPE	VARCHAR2(25 BYTE)

```
/* CREATING TABLE FOR COVEREDY_BY */
CREATE TABLE COVERED_BY
(
CLIENT_SSN NUMBER(20,0),
POLICY_ID NUMBER(20,0),
PRIMARY KEY(CLIENT_SSN,POLICY_ID),
FOREIGN KEY(CLIENT_SSN) REFERENCES CLIENTS ON DELETE CASCADE,
FOREIGN KEY(POLICY_ID) REFERENCES INSURANCE_POLICY ON DELETE CASCADE
);
```



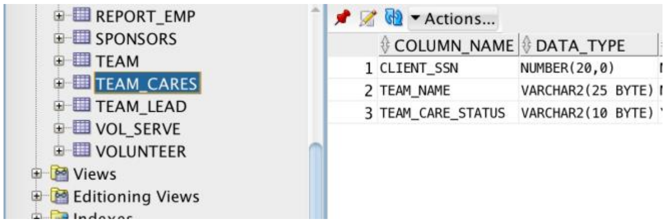
	COLUMN_NAME	DATA_TYPE
1	CLIENT_SSN	NUMBER(20,0)
2	POLICY_ID	NUMBER(20,0)

```
/* CREATING TABLE FOR PERSON */
CREATE TABLE PERSON
(
TEAM_NAME VARCHAR2(25) PRIMARY KEY,
TEAM_TYPE VARCHAR2(25),
TEAM_FORM_DATE DATE
);
```



	COLUMN_NAME	DATA_TYPE
1	TEAM_NAME	VARCHAR2(25 BYTE)
2	TEAM_TYPE	VARCHAR2(25 BYTE)
3	TEAM_FORM_DATE	DATE

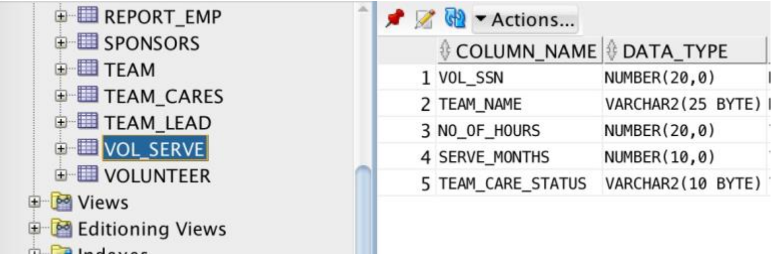
```
/* CREATING TABLE FOR TEAM_CARES */
CREATE TABLE TEAM_CARES
(
CLIENT_SSN NUMBER(20,0),
TEAM_NAME VARCHAR2(25),
TEAM_CARE_STATUS VARCHAR2(10),
PRIMARY KEY(CLIENT_SSN,TEAM_NAME),
FOREIGN KEY(CLIENT_SSN) REFERENCES CLIENTS ON DELETE CASCADE,
FOREIGN KEY(TEAM_NAME) REFERENCES TEAM ON DELETE CASCADE
);
```



	COLUMN_NAME	DATA_TYPE
1	CLIENT_SSN	NUMBER(20,0)
2	TEAM_NAME	VARCHAR2(25 BYTE)
3	TEAM_CARE_STATUS	VARCHAR2(10 BYTE)

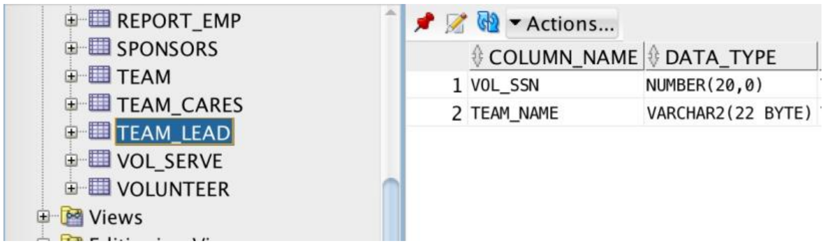
```
/* CREATING TABLE FOR VOL_SERVE */
CREATE TABLE VOL_SERVE
```

```
(
VOL_SSN  NUMBER(20,0),
TEAM_NAME VARCHAR2(25),
NO_OF_HOURS NUMBER(20,0),
SERVE_MONTHS NUMBER(10,0),
TEAM_CARE_STATUS VARCHAR(10),
PRIMARY KEY(VOL_SSN,TEAM_NAME),
FOREIGN KEY(VOL_SSN) REFERENCES VOLUNTEER ON DELETE CASCADE,
FOREIGN KEY(TEAM_NAME) REFERENCES TEAM ON DELETE CASCADE
);
```



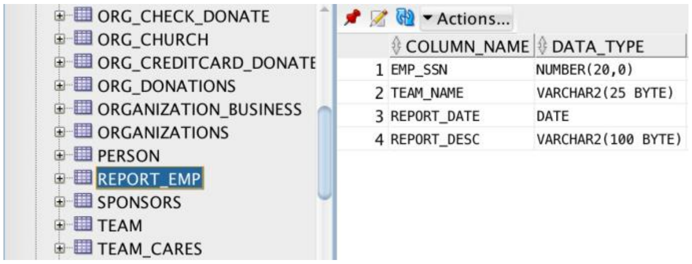
	COLUMN_NAME	DATA_TYPE
1	VOL_SSN	NUMBER(20,0)
2	TEAM_NAME	VARCHAR2(25 BYTE)
3	NO_OF_HOURS	NUMBER(20,0)
4	SERVE_MONTHS	NUMBER(10,0)
5	TEAM_CARE_STATUS	VARCHAR2(10 BYTE)

```
/* CREATING TABLE FOR TEAM_LEAD */
CREATE TABLE TEAM_LEAD
(
VOL_SSN NUMBER(20,0),
TEAM_NAME VARCHAR2(22)
);
```



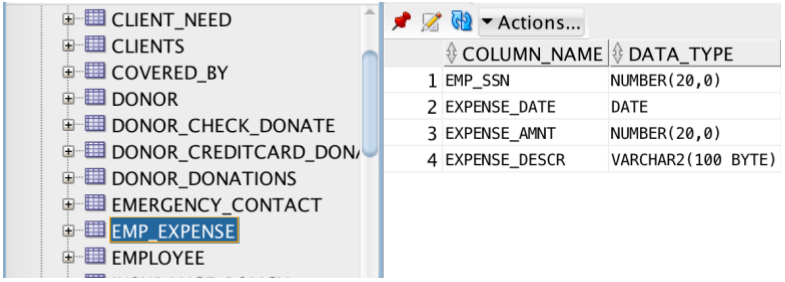
	COLUMN_NAME	DATA_TYPE
1	VOL_SSN	NUMBER(20,0)
2	TEAM_NAME	VARCHAR2(22 BYTE)

```
/* CREATING TABLE FOR REPORT_EMP */
CREATE TABLE REPORT_EMP
(
EMP_SSN NUMBER(20,0),
TEAM_NAME VARCHAR2(25),
REPORT_DATE DATE,
REPORT_DESC VARCHAR2(100),
PRIMARY KEY(EMP_SSN,TEAM_NAME),
FOREIGN KEY(EMP_SSN) REFERENCES EMPLOYEE ON DELETE CASCADE,
FOREIGN KEY(TEAM_NAME) REFERENCES TEAM ON DELETE CASCADE
);
```



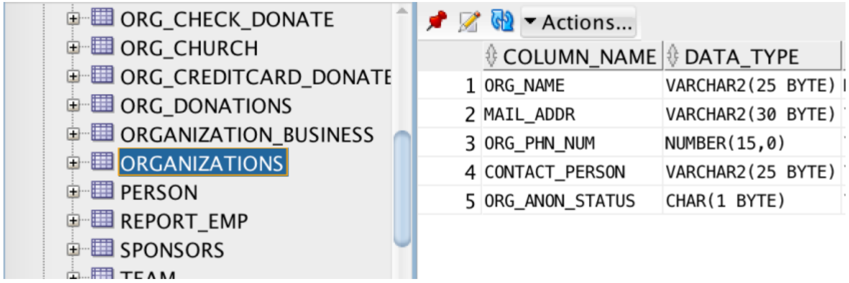
	COLUMN_NAME	DATA_TYPE
1	EMP_SSN	NUMBER(20,0)
2	TEAM_NAME	VARCHAR2(25 BYTE)
3	REPORT_DATE	DATE
4	REPORT_DESC	VARCHAR2(100 BYTE)

```
/* CREATING TABLE FOR EMP_EXPENSE */
CREATE TABLE EMP_EXPENSE
(
EMP_SSN NUMBER(20,0),
EXPENSE_DATE DATE,
EXPENSE_AMNT NUMBER(20,0),
EXPENSE_DESCR VARCHAR2(100),
PRIMARY KEY(EMP_SSN,EXPENSE_DATE,EXPENSE_AMNT,EXPENSE_DESCR),
FOREIGN KEY(EMP_SSN) REFERENCES EMPLOYEE ON DELETE CASCADE
);
```



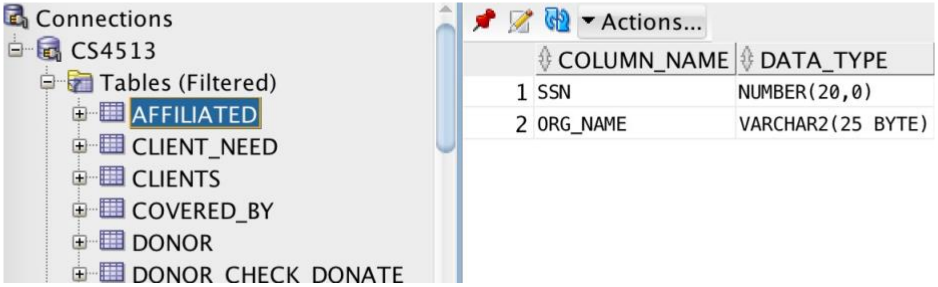
	COLUMN_NAME	DATA_TYPE
1	EMP_SSN	NUMBER(20,0)
2	EXPENSE_DATE	DATE
3	EXPENSE_AMNT	NUMBER(20,0)
4	EXPENSE_DESCR	VARCHAR2(100 BYTE)

```
/* CREATING TABLE FOR ORGANIZATIONS */
CREATE TABLE ORGANIZATIONS
(
ORG_NAME VARCHAR2(25) PRIMARY KEY,
MAIL_ADDR VARCHAR2(30),
ORG_PHN_NUM NUMBER(15,0),
CONTACT_PERSON VARCHAR2(25),
ORG_ANON_STATUS CHAR(1)
);
```



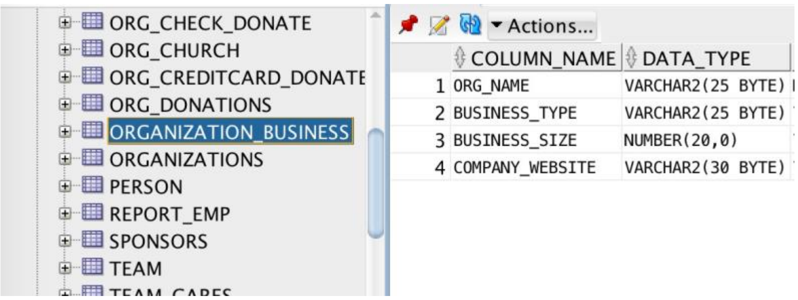
	COLUMN_NAME	DATA_TYPE
1	ORG_NAME	VARCHAR2(25 BYTE)
2	MAIL_ADDR	VARCHAR2(30 BYTE)
3	ORG_PHN_NUM	NUMBER(15,0)
4	CONTACT_PERSON	VARCHAR2(25 BYTE)
5	ORG_ANON_STATUS	CHAR(1 BYTE)

```
/* CREATING TABLE FOR AFFILIATED */
CREATE TABLE AFFILIATED
(
SSN  NUMBER(20,0),
ORG_NAME VARCHAR2(25),
PRIMARY KEY(SSN,ORG_NAME),
FOREIGN KEY(SSN) REFERENCES PERSON ON DELETE CASCADE,
FOREIGN KEY(ORG_NAME) REFERENCES ORGANIZATIONS ON DELETE CASCADE
);
```



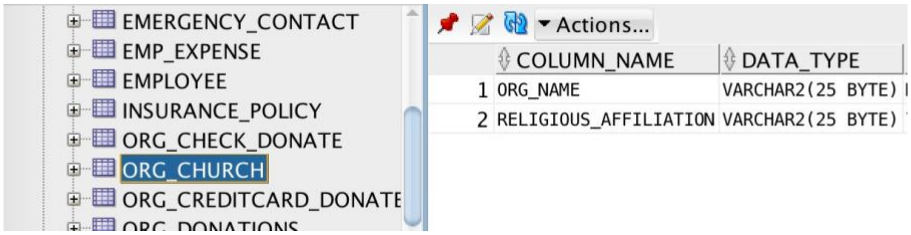
	COLUMN_NAME	DATA_TYPE
1	SSN	NUMBER(20,0)
2	ORG_NAME	VARCHAR2(25 BYTE)

```
/* CREATING TABLE FOR ORGANIZATION_BUSINESS */
CREATE TABLE ORGANIZATION_BUSINESS
(
ORG_NAME VARCHAR2(25) PRIMARY KEY,
BUSINESS_TYPE VARCHAR2(25),
BUSINESS_SIZE NUMBER(20,0),
COMPANY_WEBSITE VARCHAR2(30),
FOREIGN KEY(ORG_NAME) REFERENCES ORGANIZATIONS ON DELETE CASCADE
);
```



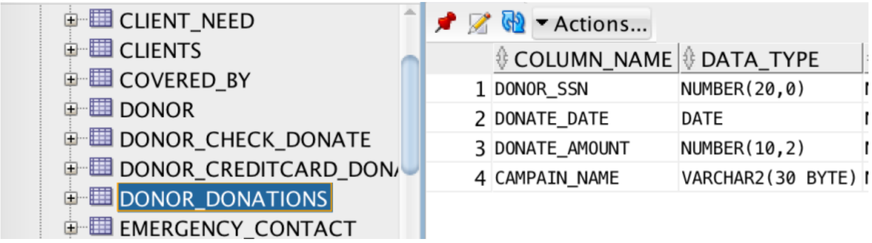
	COLUMN_NAME	DATA_TYPE
1	ORG_NAME	VARCHAR2(25 BYTE)
2	BUSINESS_TYPE	VARCHAR2(25 BYTE)
3	BUSINESS_SIZE	NUMBER(20,0)
4	COMPANY_WEBSITE	VARCHAR2(30 BYTE)

```
/* CREATING TABLE FOR ORG_CHURCH */
CREATE TABLE ORG_CHURCH
(
ORG_NAME VARCHAR2(25) PRIMARY KEY,
RELIGIOUS_AFFILIATION VARCHAR2(25),
FOREIGN KEY(ORG_NAME) REFERENCES ORGANIZATIONS ON DELETE CASCADE
);
```



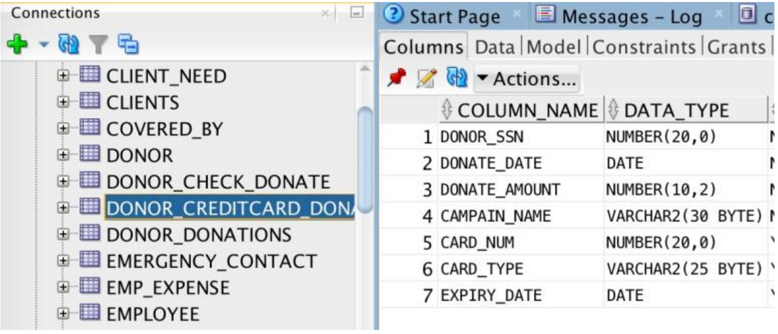
	COLUMN_NAME	DATA_TYPE
1	ORG_NAME	VARCHAR2(25 BYTE)
2	RELIGIOUS_AFFILIATION	VARCHAR2(25 BYTE)

```
/* CREATING TABLE FOR DONOR_DONATIONS */
CREATE TABLE DONOR_DONATIONS
(
DONOR_SSN NUMBER(20,0),
DONATE_DATE DATE,
DONATE_AMOUNT NUMBER(10,2),
CAMPAIN_NAME VARCHAR2(30),
PRIMARY KEY(DONOR_SSN,DONATE_DATE,DONATE_AMOUNT,CAMPAIN_NAME),
FOREIGN KEY(DONOR_SSN) REFERENCES DONOR ON DELETE CASCADE
);
```



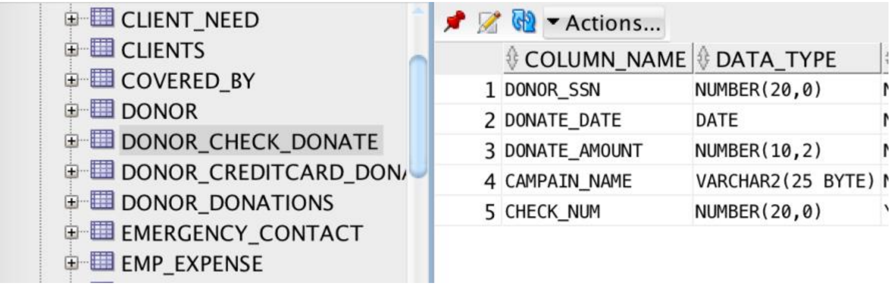
	COLUMN_NAME	DATA_TYPE
1	DONOR_SSN	NUMBER(20,0)
2	DONATE_DATE	DATE
3	DONATE_AMOUNT	NUMBER(10,2)
4	CAMPAIN_NAME	VARCHAR2(30 BYTE)

```
/* CREATING TABLE FOR DONOR_CREDITCARD_DONATE */
CREATE TABLE DONOR_CREDITCARD_DONATE
(
DONOR_SSN NUMBER(20,0),
DONATE_DATE DATE,
DONATE_AMOUNT NUMBER(10,2),
CAMPAIN_NAME VARCHAR2(30),
CARD_NUM NUMBER(20,0),
CARD_TYPE VARCHAR2(25),
EXPIRY_DATE DATE,
PRIMARY KEY(DONOR_SSN,DONATE_DATE,DONATE_AMOUNT,CAMPAIN_NAME),
FOREIGN KEY(DONOR_SSN,DONATE_DATE,DONATE_AMOUNT,CAMPAIN_NAME) REFERENCES
DONOR_DONATIONS ON DELETE CASCADE
);
```



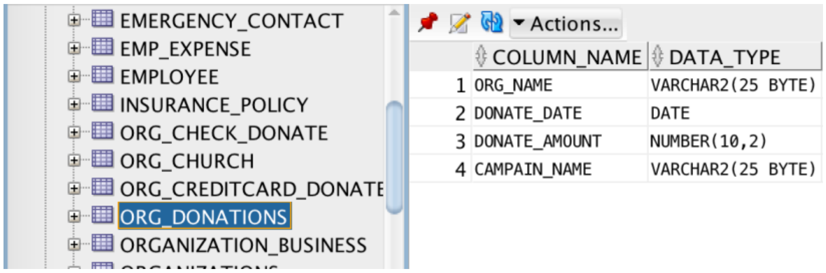
	COLUMN_NAME	DATA_TYPE
1	DONOR_SSN	NUMBER(20,0)
2	DONATE_DATE	DATE
3	DONATE_AMOUNT	NUMBER(10,2)
4	CAMPAIN_NAME	VARCHAR2(30 BYTE)
5	CARD_NUM	NUMBER(20,0)
6	CARD_TYPE	VARCHAR2(25 BYTE)
7	EXPIRY_DATE	DATE

```
/* CREATING TABLE FOR CHECK_DONATE */
CREATE TABLE DONOR_CHECK_DONATE
(
DONOR_SSN NUMBER(20,0),
DONATE_DATE DATE,
DONATE_AMOUNT NUMBER(10,2),
CAMPAIN_NAME VARCHAR2(25),
CHECK_NUM NUMBER(20,0),
PRIMARY KEY(DONOR_SSN,DONATE_DATE,DONATE_AMOUNT,CAMPAIN_NAME),
FOREIGN KEY(DONOR_SSN,DONATE_DATE,DONATE_AMOUNT,CAMPAIN_NAME) REFERENCES
DONOR_DONATIONS ON DELETE CASCADE
);
```



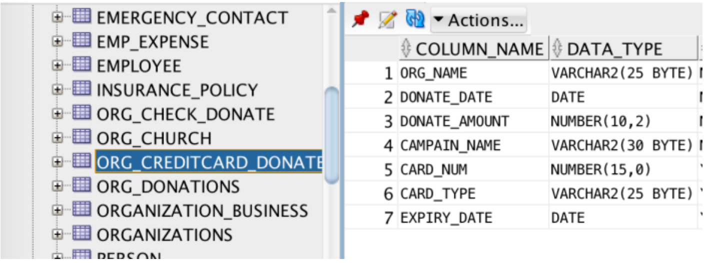
	COLUMN_NAME	DATA_TYPE
1	DONOR_SSN	NUMBER(20,0)
2	DONATE_DATE	DATE
3	DONATE_AMOUNT	NUMBER(10,2)
4	CAMPAIN_NAME	VARCHAR2(25 BYTE)
5	CHECK_NUM	NUMBER(20,0)

```
/* CREATING TABLE FOR ORG_DONATIONS */
CREATE TABLE ORG_DONATIONS
(
ORG_NAME VARCHAR2(25),
DONATE_DATE DATE,
DONATE_AMOUNT NUMBER(10,2),
CAMPAIN_NAME VARCHAR2(25),
PRIMARY KEY(ORG_NAME,DONATE_DATE,DONATE_AMOUNT,CAMPAIN_NAME),
FOREIGN KEY(ORG_NAME) REFERENCES ORGANIZATIONS ON DELETE CASCADE
);
```



	COLUMN_NAME	DATA_TYPE
1	ORG_NAME	VARCHAR2(25 BYTE)
2	DONATE_DATE	DATE
3	DONATE_AMOUNT	NUMBER(10,2)
4	CAMPAIN_NAME	VARCHAR2(25 BYTE)

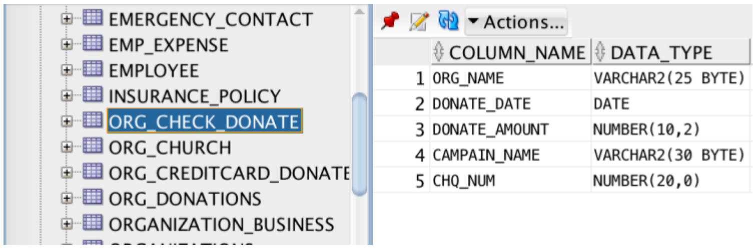
```
/* CREATING TABLE FOR ORG_CREDITCARD_DONATE */
CREATE TABLE ORG_CREDITCARD_DONATE
(
ORG_NAME VARCHAR2(25),
DONATE_DATE DATE,
DONATE_AMOUNT NUMBER(10,2),
CAMPAIN_NAME VARCHAR2(30),
CARD_NUM NUMBER(15,0),
CARD_TYPE VARCHAR2(25),
EXPIRY_DATE DATE,
PRIMARY KEY(ORG_NAME,DONATE_DATE,DONATE_AMOUNT,CAMPAIN_NAME),
FOREIGN KEY(ORG_NAME,DONATE_DATE,DONATE_AMOUNT,CAMPAIN_NAME) REFERENCES ORG_DONATIONS
ON DELETE CASCADE
);
```



EMERGENCY_CONTACT
EMP_EXPENSE
EMPLOYEE
INSURANCE_POLICY
ORG_CHECK_DONATE
ORG_CHURCH
ORG_CREDITCARD_DONATE
ORG_DONATIONS
ORGANIZATION_BUSINESS
ORGANIZATIONS
PERSON

	COLUMN_NAME	DATA_TYPE
1	ORG_NAME	VARCHAR2(25 BYTE)
2	DONATE_DATE	DATE
3	DONATE_AMOUNT	NUMBER(10,2)
4	CAMPAIN_NAME	VARCHAR2(30 BYTE)
5	CARD_NUM	NUMBER(15,0)
6	CARD_TYPE	VARCHAR2(25 BYTE)
7	EXPIRY_DATE	DATE

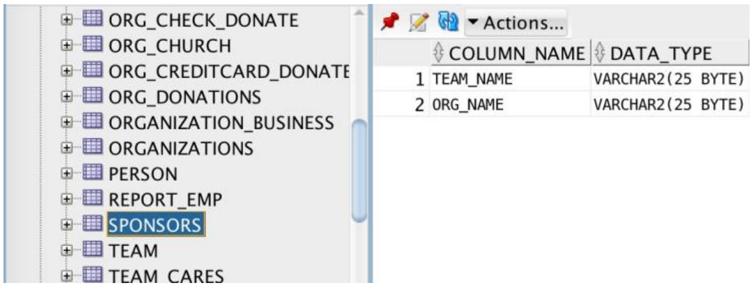
```
/* CREATING TABLE FOR ORG_CHECK_DONATE */
CREATE TABLE ORG_CHECK_DONATE
(
ORG_NAME VARCHAR2(25),
DONATE_DATE DATE,
DONATE_AMOUNT NUMBER(10,2),
CAMPAIN_NAME VARCHAR2(30),
CHQ_NUM NUMBER(20,0),
PRIMARY KEY(ORG_NAME,DONATE_DATE,DONATE_AMOUNT,CAMPAIN_NAME),
FOREIGN KEY(ORG_NAME,DONATE_DATE,DONATE_AMOUNT,CAMPAIN_NAME) REFERENCES ORG_DONATIONS
ON DELETE CASCADE
);
```



EMERGENCY_CONTACT
EMP_EXPENSE
EMPLOYEE
INSURANCE_POLICY
ORG_CHECK_DONATE
ORG_CHURCH
ORG_CREDITCARD_DONATE
ORG_DONATIONS
ORGANIZATION_BUSINESS

	COLUMN_NAME	DATA_TYPE
1	ORG_NAME	VARCHAR2(25 BYTE)
2	DONATE_DATE	DATE
3	DONATE_AMOUNT	NUMBER(10,2)
4	CAMPAIN_NAME	VARCHAR2(30 BYTE)
5	CHQ_NUM	NUMBER(20,0)

```
/* CREATING TABLE FOR SPONSORS */
CREATE TABLE SPONSORS
(
TEAM_NAME VARCHAR2(25),
ORG_NAME VARCHAR2(25),
PRIMARY KEY(TEAM_NAME,ORG_NAME),
FOREIGN KEY(TEAM_NAME) REFERENCES TEAM ON DELETE CASCADE,
FOREIGN KEY(ORG_NAME) REFERENCES ORGANIZATIONS ON DELETE CASCADE
);
```



	COLUMN_NAME	DATA_TYPE
1	TEAM_NAME	VARCHAR2(25 BYTE)
2	ORG_NAME	VARCHAR2(25 BYTE)

INSERT STATEMENTS:

```
/*PERSON TABLE*/
INSERT INTO PERSON VALUES('SUMITH',1001,'27-MAR-
1991','INDIAN','MALE','SOFTWARE','NORMAN','SUMITH@GMAIL.COM','Y',4054642839,7838755594,7875759494);
INSERT INTO PERSON VALUES('JUUHWA',2001,'27-APR-
1990','JAPAN','FEMALE','CEO','AUSTIN','JUUHWA@GMAIL.COM','N',512890100,NULL,NULL);
INSERT INTO PERSON VALUES('KICE',3001,'27-MAY-1989','CHINA','MALE','VICE
PRESIDENT','DALLAS','KICE@GMAIL.COM','Y',NULL,214999222,NULL);
INSERT INTO PERSON VALUES('MAHESH',4001,'27-JUN-
1988','INDIAN','MALE','SOFTWARE','BOSTON','MAHESH@GMAIL.COM','N',812456345,NULL,9848062228);
INSERT INTO PERSON VALUES('KYU',5001,'27-JUL-
1990','CHINA','FEMALE','LAWYER','NORMAN','KYU@GMAIL.COM','Y',NULL,405456654,NULL);
INSERT INTO PERSON VALUES('CHARLES',6001,'27-AUG-
1986','AMERICAN','MALE','PROFESSOR','OKLAHOMA','CHARLES@GMAIL.COM','Y',NULL,405909890,NULL);
INSERT INTO PERSON VALUES('GRANT',7001,'27-SEP-
1985','AMERICAN','MALE','PROFESSOR','TULSA','GRANT@GMAIL.COM','Y',NULL,4054642830,917345321);
INSERT INTO PERSON VALUES('RUSHI',8001,'27-OCT-
1991','INDIAN','MALE','ENGINEER','INDIA','RUSHI@GMAIL.COM','Y',8897595656,9848022338,9921345222);
INSERT INTO PERSON VALUES('ARPIT',9001,'27-NOV-
1991','INDIAN','MALE','SOFTWARE','INDIA','ARPIT@GMAIL.COM','Y',NULL,NULL,123456789);
INSERT INTO PERSON VALUES('VAIBHAV',1101,'27-DEC-
1991','INDIAN','MALE','SOFTWARE','INDIA','VAIBHAV@GMAIL.COM','Y',NULL,987654321,NULL);
INSERT INTO PERSON VALUES('RUSHI',1201,'27-JAN-
1991','INDIAN','MALE','ENGINEER','INDIA','RUSHI@GMAIL.COM','Y',8897595656,9848022338,9921345222);
INSERT INTO PERSON VALUES('ARPIT',1301,'27-FEB-
1991','INDIAN','MALE','SOFTWARE','INDIA','ARPT@GMAIL.COM','Y',NULL,NULL,123456789);
INSERT INTO PERSON VALUES('VAIBHAV',1401,'1-APR-
1991','INDIAN','MALE','SOFTWARE','INDIA','VAIBHAV@GMAIL.COM','Y',NULL,987654321,NULL);
INSERT INTO PERSON VALUES('SATISH',1501,'27-DEC-
1993','INDIAN','MALE','SOFTWARE','INDIA','SATISH@GMAIL.COM','Y',NULL,988794321,NULL);
INSERT INTO PERSON VALUES('LINDA',1601,'27-JAN-
1993','AMERICAN','FEMALE','ENGINEER','USA','LINDA@GMAIL.COM','Y',8897595456,NULL,9555345222);
INSERT INTO PERSON VALUES('LISA',1701,'27-FEB-
1993','AMERICAN','FEMALE','PCC','USA','LISA@GMAIL.COM','Y',NULL,NULL,9887747474);
INSERT INTO PERSON VALUES('DAVID',1801,'1-APR-
1993','INDIAN','MALE','SOFTWARE','INDIA','DAVID@GMAIL.COM','Y',NULL,9876234321,NULL);

/* EMERGENCY_CONTACT */
INSERT INTO EMERGENCY_CONTACT VALUES(1001,'AVINASH',9030217545,'FRIEND');
INSERTINTOEMERGENCY_CONTACTVALUES(2001,'SRIDHAR',4054842839,'BROTHER');
INSERT INTO EMERGENCY_CONTACT VALUES(7001,'VARAHAN',4053251954,'COUSIN');
INSERT INTO EMERGENCY_CONTACT VALUES(4001,'KARTHIK',9341223432,'BEST FRIEND');
INSERT INTO EMERGENCY_CONTACT VALUES(6001,'MOHEBBI',4052643453,'SISTER');

/* CLIENTS */
INSERT INTO CLIENTS VALUES(1001,'SHETTY',6030217545,'RAJKISHORE',8030267645,'1-JAN-1991');
INSERT INTO CLIENTS VALUES(3001,'ADAMS',7054842839,'AGARWAL',6030217321,'2-FEB-1992');
INSERT INTO CLIENTS VALUES(4001,'LARRY',2053251954,'AGNIHOTRI',1230217545,'3-MAR-1993');
INSERT INTO CLIENTS VALUES(7001,'DEMENT',8341223432,'UMESH',3210217545,'4-APR-1994');
INSERT INTO CLIENTS VALUES(5001,'WILLIAM',2052643457,'GUDURU',5430217545,'5-MAY-1995');

/* VOLUNTEER */
INSERT INTO VOLUNTEER VALUES(1001,'1-JAN-1991','15-JAN-1991','OKLAHOMA');
INSERT INTO VOLUNTEER VALUES(8001,'2-FEB-1992','15-FEB-1992','PUNE'); INSERT
INTO VOLUNTEER VALUES(9001,'3-MAR-1993','15-MAR-1993','NOIDA'); INSERT
INTO VOLUNTEER VALUES(1101,'4-APR-1994','15-APR-1994','NOIDA'); INSERT INTO
VOLUNTEER VALUES(5001,'5-MAY-1995','15-MAY-1995','NORMAN');

/* EMPLOYEE */
INSERT INTO EMPLOYEE VALUES(3001,750000,'SINGLE','1-MAR-1991');
INSERT INTO EMPLOYEE VALUES(4001,800000,'SINGLE','2-APR-1991');
INSERT INTO EMPLOYEE VALUES(8001,850000,'MARRIED','3-AUG-1991');
INSERT INTO EMPLOYEE VALUES(9001,900000,'MARRIED','9-SEP-1991');
INSERT INTO EMPLOYEE VALUES(1101,700000,'MARRIED','10-OCT-1991');

/* DONOR */
INSERT INTO DONOR VALUES(1001,'Y');
INSERT INTO DONOR VALUES(2001,'N');
INSERT INTO DONOR VALUES(3001,'N');
INSERT INTO DONOR VALUES(4001,'Y');
INSERT INTO DONOR VALUES(5001,'N');

/* CLIENT_NEED */
INSERT INTO CLIENT_NEED VALUES(1001,'SLEEP',10);
INSERT INTO CLIENT_NEED VALUES(3001,'STUDY',6);
INSERT INTO CLIENT_NEED VALUES(4001,'TRANSPORTATION',5);
INSERT INTO CLIENT_NEED VALUES(5001,'FOOD',10);
INSERT INTO CLIENT_NEED VALUES(5001,'TRANSPORTATION',8);
INSERT INTO CLIENT_NEED VALUES(3001,'TRANSPORTATION',4);
INSERT INTO CLIENT_NEED VALUES(3001,'TRANSPORTATION',3);

/* INSURANCE_POLICY */
INSERT INTO INSURANCE_POLICY VALUES(9999,1234,'NORMAN','HEALTH');
INSERT INTO INSURANCE_POLICY VALUES(8888,5678,'OKLAHOMA','LIFE INSURANCE');
INSERT INTO INSURANCE_POLICY VALUES(7777,9123,'TULSA','HEALTH');
INSERT INTO INSURANCE_POLICY VALUES(6666,9876,'DALLAS','DENTAL');
INSERT INTO INSURANCE_POLICY VALUES(5555,3456,'DENVER','LIFE INSURANCE');


/* COVERED_BY */
INSERT INTO COVERED_BY VALUES(1001,9999);
INSERT INTO COVERED_BY VALUES(3001,7777);
INSERT INTO COVERED_BY VALUES(4001,6666);
INSERT INTO COVERED_BY VALUES(5001,5555);
```

```
/* TEAM */
INSERT INTO TEAM VALUES('TUS14','TYPE 1','31-MAR-1991');
INSERT INTO TEAM VALUES('TUS16','TYPE 3','31-AUG-1991');
INSERT INTO TEAM VALUES('IMS01','TYPE 5','31-OCT-1991');

/* TEAM_CARES */
INSERT INTO TEAM_CARES VALUES(1001,'TUS14','ACTIVE');
INSERT INTO TEAM_CARES VALUES(4001,'TUS16','ACTIVE');
INSERT INTO TEAM_CARES VALUES(5001,'IMS01','INACTIVE');

/* VOL_SERVE */
INSERT INTO VOL_SERVE VALUES(1001,'TUS14',20,3,'INACTIVE');
INSERT INTO VOL_SERVE VALUES(1001,'TUS16',20,4,'INACTIVE');
INSERT INTO VOL_SERVE VALUES(1101,'IMS01',15,9,'ACTIVE');
INSERT INTO VOL_SERVE VALUES(8001,'TUS14',20,3,'INACTIVE');
INSERT INTO VOL_SERVE VALUES(8001,'TUS16',40,4,'INACTIVE');

/* TEAM_LEAD */
INSERT INTO TEAM_LEAD VALUES(1001,'TUS14');
INSERT INTO TEAM_LEAD VALUES(8001,'IMS01');
INSERT INTO TEAM_LEAD VALUES(1101, 'TUS15');

/* REPORT_EMP */
INSERT INTO REPORT_EMP VALUES(3001,'IMS01','31-MAR-1992','PROJECT 1 COMPLETED');
INSERT INTO REPORT_EMP VALUES(4001,'TUS16','30-APR-1992','TASK 1 COMPLETED');
INSERT INTO REPORT_EMP VALUES(8001,'IMS01','30-AUG-1992','PROJECT 2 COMPLETED');
INSERT INTO REPORT_EMP VALUES(3001,'TUS16','30-APR-1992','TASK 2 COMPLETED');
INSERT INTO REPORT_EMP VALUES(4001,'IMS01','30-AUG-1992','PROJECT 5 COMPLETED');

/* EMP_EXPENSE */
INSERT INTO EMP_EXPENSE VALUES(3001,'1-APR-1991',10000,'TRAVEL EXPENSES'); INSERT
INTO EMP_EXPENSE VALUES(4001,'2-MAY-1992',20000,'FOOD EXPENSES');  INSERT INTO
EMP_EXPENSE VALUES(8001,'3-NOV-1993',30000,'TRAVEL EXPENSES'); INSERT INTO
EMP_EXPENSE VALUES(9001,'4-DEC-1994',10000,'ONE TIME ALLOWANCES');

/* ORGANIZATIONS */
INSERT INTO ORGANIZATIONS VALUES('GOOGLE','San Fransisco',405123321,'SUMITH','N');
INSERT INTO ORGANIZATIONS VALUES('AMAZON','Chicago',980765890,'SHRANITH','Y');
INSERT INTO ORGANIZATIONS VALUES('MICROSOFT','Washington',9847840984,'NAVEEN','Y');
INSERT INTO ORGANIZATIONS VALUES('APPLE','California',430321543,'PRAMOD','N');

/* AFFILIATED */
INSERT INTO AFFILIATED VALUES(1001,'GOOGLE');
INSERT INTO AFFILIATED VALUES(3001,'AMAZON');
INSERT INTO AFFILIATED VALUES(4001,'GOOGLE');
INSERT INTO AFFILIATED VALUES(7001,'APPLE');
INSERT INTO AFFILIATED VALUES(5001,'APPLE');

/* ORGANIZATION_BUSINESS */
INSERT INTO ORGANIZATION_BUSINESS VALUES('GOOGLE','SOFTWARE',15000,'GOOGLE.COM');
INSERT INTO ORGANIZATION_BUSINESS VALUES('AMAZON','HARDWARE',25000,'AMAZON.COM');
```

```
/* ORGANIZATION_BUSINESS */
INSERT INTO ORG_CHURCH VALUES('MICROSOFT','CHRISTIAN');
INSERT INTO ORG_CHURCH VALUES('APPLE','HINDU');

/* DONOR_DONATIONS */
INSERT INTO DONOR_DONATIONS VALUES(3001,'1-MAY-1991',9000,'COMPAIN 1');
INSERT INTO DONOR_DONATIONS VALUES(4001,'2-JUNE-1992',60000,'COMPAIN 2');
INSERT INTO DONOR_DONATIONS VALUES(1001,'3-DEC-1993',20000,'COMPAIN 3');
INSERT INTO DONOR_DONATIONS VALUES(5001,'4-DEC-1994',90000,'COMPAIN 4');

/* DONOR_CREDITCARD_DONATE */
INSERT INTO DONOR_CREDITCARD_DONATE VALUES(3001,'1-MAY-1991',9000,'COMPAIN
1',19001888876541234,'VISA','24-DEC-2018');
INSERT INTO DONOR_CREDITCARD_DONATE VALUES(4001,'2-JUNE-1992',60000,'COMPAIN
2',0980675402022222,'MASTER','12-DEC-2019');
INSERT INTO DONOR_CREDITCARD_DONATE VALUES(1001,'3-DEC-1993',20000,'COMPAIN
3',897822233355555,'VISA','11-DEC-2017');
INSERT INTO DONOR_CREDITCARD_DONATE VALUES(5001,'4-DEC-1994',90000,'COMPAIN
4',987345762330208,'DISCOVER','15-DEC-2016');

/* DONOR_CHECK_DONATE */
INSERT INTO DONOR_CHECK_DONATE VALUES(3001,'1-MAY-1991',9000,'COMPAIN 1',111);
INSERT INTO DONOR_CHECK_DONATE VALUES(4001,'2-JUNE-1992',60000,'COMPAIN 2',222);
INSERT INTO DONOR_CHECK_DONATE VALUES(1001,'3-DEC-1993',20000,'COMPAIN 3',333);
INSERT INTO DONOR_CHECK_DONATE VALUES(5001,'4-DEC-1994',90000,'COMPAIN 4',444);

/* ORG_DONATIONS */
INSERT INTO ORG_DONATIONS VALUES('GOOGLE','1-MAY-1995',100000,'ORG COMPAIN 1');
INSERT INTO ORG_DONATIONS VALUES('GOOGLE','2-JUNE-1995',400000,'ORG COMPAIN 2');
INSERT INTO ORG_DONATIONS VALUES('MICROSOFT','3-DEC-1995',600000,'ORG COMPAIN 3');
INSERT INTO ORG_DONATIONS VALUES('APPLE','4-DEC-1995',950000,'ORG COMPAIN 4');

/* ORG_CREDITCARD_DONATE */
INSERT INTO ORG_CREDITCARD_DONATE VALUES('GOOGLE','1-MAY-1995',100000,'ORG COMPAIN
1',123234345456,'VISA','24-MAR-2018');
INSERT INTO ORG_CREDITCARD_DONATE VALUES('GOOGLE','2-JUNE-1995',400000,'ORG COMPAIN
2',345456456767,'VISA','15-APR-2019');
INSERT INTO ORG_CREDITCARD_DONATE VALUES('MICROSOFT','3-DEC-1995',600000,'ORG COMPAIN
3',456567678789,'DISCOVER','15-MAR-2019');
INSERT INTO ORG_CREDITCARD_DONATE VALUES('APPLE','4-DEC-1995',950000,'ORG COMPAIN
4',234345456567,'MASTER','20-AUG-2019');

/* ORG_CREDITCARD_DONATE */
INSERT INTO ORG_CHECK_DONATE VALUES('GOOGLE','1-MAY-1995',100000,'ORG COMPAIN 1',123);
INSERT INTO ORG_CHECK_DONATE VALUES('GOOGLE','2-JUNE-1995',400000,'ORG COMPAIN 2',456);
INSERT INTO ORG_CHECK_DONATE VALUES('MICROSOFT','3-DEC-1995',600000,'ORG COMPAIN 3',789);
INSERT INTO ORG_CHECK_DONATE VALUES('APPLE','4-DEC-1995',950000,'ORG COMPAIN 4',001);

/* SPONSORS */
INSERT INTO SPONSORS VALUES('TUS14','GOOGLE');
INSERT INTO SPONSORS VALUES('TUS16','AMAZON');
INSERT INTO SPONSORS VALUES('TUS16','GOOGLE');
INSERT INTO SPONSORS VALUES('IMS01','APPLE');
```

TASK4: JAVA APPLICATION PROGRAM:

```
/**
 * @author Sumith Kumar Gannarapu
 *
 */
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.DataInputStream;
import java.io.FileInputStream;
import java.io.FileWriter;
import java.io.InputStreamReader;
import java.sql.*;

public class IP_JAVA_GANNARAPU_SUMITH {
 public static void main(String[] args) throws ClassNotFoundException, SQLException {
  //Loading a database driver
  try {
   Class.forName("oracle.jdbc.driver.OracleDriver");
  } catch (Exception e) {
   System.out.println("Unable to load the driver class");
  }
  Connection conn = null;
  //Creating an Oracle JDBC Connection.
  try {

   System.out.println("Connecting to database...");
   conn =
DriverManager.getConnection("jdbc:oracle:thin:@//oracle.cs.ou.edu:1521/pdborcl.cs.ou.edu",
"gann0001", "DSlt3Dt5");
   System.out.println("connection established");
   //Creating a JDBC Statement object
   Statement st = conn.createStatement();
   BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
   String query;
   String query1;
   System.out.println("WELCOME TO PAN CLIENT AND DONOR DATABASE SYSTEM");
   while (true) {
    System.out.println("-------------------------------------- OPTIONS --------------------
---------------");
    System.out.println("1.  Enter a new team into the database");
    System.out.println("2.  Enter a new client into the database and associate him or her
with one or more teams");
    System.out.println("3.  Enter a new volunteer into the database and associate him or
her with one or more teams");
    System.out.println("4.  Enter the number of hours a volunteer worked this month for a
particular team");
    System.out.println("5.  Enter a new employee into the database and associate him or her
with one or more teams");
    System.out.println("6.  Enter an expense charged by an employee");
    System.out.println("7.  Enter a new organization and associate it to one or more PAN
teams");
    System.out.println("8.  Enter a new donor and associate him or her with several
donations");
    System.out.println("9.  Enter a new organization and associate it with several
donations");
```

```
    System.out.println("10. Retrieve the name and phone number of the doctor of a
particular client");
    System.out.println("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");
    System.out.println("12. Retrieve the list of volunteers that are members of teams that
support a particular client");
    System.out.println("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");
    System.out.println("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");
    System.out.println("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");
    System.out.println("16. Increase the salary by 10% of all employees to whom more than
one team must report");
    System.out.println("17. Delete all clients who do not have health insurance and whose
value of importance for transportation is less than 5");
    System.out.println("18. Import: Enter new teams from a data file until the file is
empty");
    System.out.println("19. Export: Retrieve names and mailing addresses of all people on
the mailing list and output them to a data file");
    System.out.println("20. Quit");
    System.out.println("Enter Your Choice");
    int choice = Integer.parseInt(br.readLine());
    switch (choice) {
     case 1:
      //Enter a new team into the database
      // Storing Values in variables
      System.out.println("Enter Team Name:");
      String team_name = br.readLine();
      System.out.println("Enter Team Type:");
      String team_type = br.readLine();
      System.out.println("Enter Team Form Date - Please enter date in the form of 01-JAN-
2017:");
      String team_form_date = br.readLine();
      query = "Insert into Team values('" + team_name + "','" + team_type + "','" +
team_form_date + "')";
      st.executeUpdate(query);
      System.out.println("Database records for team table has been inserted");
      break;

     case 2:
      //Enter a new client into the database and associate him or her with one or more
teams
      // Storing Values in variables
      System.out.println("Executing Query 2 ...");
      System.out.println("Enter Client SSN:");
      int client_ssn = Integer.parseInt(br.readLine());
      System.out.println("Enter Doctor Name:");
      String client_doc_name = br.readLine();
      System.out.println("Enter Doctor Phone Number:");
      long client_doc_phn = Long.parseLong(br.readLine());
      System.out.println("Enter Attorney Name:");
      String client_attorney_name = br.readLine();
      System.out.println("Enter Attorney Phone Number:");
      long client_attorney_phn = Long.parseLong(br.readLine());
      System.out.println("Enter Client Assign Date - Please enter date in the form of 01-
JAN-2017:");
      String client_assign_date = br.readLine();
      query = "Insert into clients values(" + client_ssn + ",'" + client_doc_name + "','" +
client_doc_phn + "','" + client_attorney_name + "','" + client_attorney_phn + "','" +
client_assign_date + "')";
      st.executeUpdate(query);
      System.out.println("How many teams you want to associate\t" + client_ssn + "\tclient
?");
      int num = Integer.parseInt(br.readLine());
      for (int i = 1; i <= num; i++) {
       System.out.println("Enter Team Name:");
       team_name = br.readLine();
       System.out.println("Enter Team Care Status:");
       String team_care_status = br.readLine();
       query1 = "insert into team_cares values(" + client_ssn + ",'" + team_name + "','" +
team_care_status + "')";
       st.executeUpdate(query1);
      }
      System.out.println("Database records for clients table has been inserted and" +
client_ssn + "associated with one or more teams");
      break;

     case 3:
      //Enter a new volunteer into the database and associate him or her with one or more
teams
      // Storing Values in variables
      System.out.println("Enter Volunteer SSN:");
      int vol_ssn = Integer.parseInt(br.readLine());
      System.out.println("Enter Vol Join Date - Please enter date in the form of 01-JAN-
2017:");
      String vol_join_date = br.readLine();
      System.out.println("Enter Vol Train Date - Please enter date in the form of 01-JAN-
2017:");
      String vol_train_date = br.readLine();
      System.out.println("Enter Vol Train Location:");
      String vol_train_loc = br.readLine();
      query = "Insert into volunteer values('" + vol_ssn + "','" + vol_join_date + "','" +
vol_train_date + "','" + vol_train_loc + "')";
      st.executeUpdate(query);
      System.out.println("How many teams you want to associate\t" + vol_ssn +
"\tVolunteer?");
      num = Integer.parseInt(br.readLine());
      for (int i = 1; i <= num; i++) {
       System.out.println("Enter Team Name:");
       team_name = br.readLine();
       System.out.println("Enter No Of Hours worked:");
       int no_of_hours = Integer.parseInt(br.readLine());
       System.out.println("Enter Serve Months:");
       int serve_months = Integer.parseInt(br.readLine());
       System.out.println("Enter Team Care Status:");
       String team_care_status = br.readLine();
       query1 = "insert into vol_serve values('" + vol_ssn + "','" + team_name + "','" +
no_of_hours + "','" + serve_months + "','" + team_care_status + "')";
       st.executeUpdate(query1);
      }
      System.out.println("Database records for Volunteer table has been inserted and\t" +
vol_ssn + "\tassociated with one or more teams");
      break;

     case 4:
      //Enter the number of hours a volunteer worked this month for a particular team
      // Storing Values in variables
      System.out.println("Enter Volunteer SSN:");
      vol_ssn = Integer.parseInt(br.readLine());
      System.out.println("Enter Team Name:");
      team_name = br.readLine();
      System.out.println("Enter No Of Hours:");
      int no_of_hours = Integer.parseInt(br.readLine());
      System.out.println("Enter Serve Month:");
      int serve_months = Integer.parseInt(br.readLine());
      System.out.println("Enter Team Care Status:");
      String team_care_status = br.readLine();
      query = "insert into vol_serve values('" + vol_ssn + "','" + team_name + "','" +
no_of_hours + "','" + serve_months + "','" + team_care_status + "')";
      st.executeUpdate(query);
      System.out.println("Database records for Volunteer with number of hours worked for
particular months has been inserted");
      break;

     case 5:
      //Enter a new employee into the database and associate him or her with one or more
teams
      // Storing Values in variables
      System.out.println("Enter Employee SSN:");
      int emp_ssn = Integer.parseInt(br.readLine());
      System.out.println("Enter Employee Salary:");
      int emp_sal = Integer.parseInt(br.readLine());
      System.out.println("Enter Marrital Status:");
      String emp_marrital_status = br.readLine();
      System.out.println("Enter Hire Date - Please enter date in the form of 01-JAN-
2017:");
      String emp_hire_date = br.readLine();
      query = "Insert into employee values('" + emp_ssn + "','" + emp_sal + "','" +
emp_marrital_status + "','" + emp_hire_date + "')";
      st.executeUpdate(query);
      System.out.println("How many teams you want to associate\t" + emp_ssn +
"\tEmployee?");
      num = Integer.parseInt(br.readLine());
      for (int i = 1; i <= num; i++) {
       System.out.println("Enter Team Name:");
       team_name = br.readLine();
       System.out.println("Enter Report Date - Please enter date in the form of 01-JAN-
2017:");
       String report_date = br.readLine();
       System.out.println("Enter Report Description:");
       String report_desc = br.readLine();
       query1 = "insert into report_emp values('" + emp_ssn + "','" + team_name + "','" +
report_date + "','" + report_desc + "')";
       st.executeUpdate(query1);
      }
      System.out.println("Database records for employee table has been inserted and\t" +
emp_ssn + "\tassociated with one or more teams");
```

```
      break;

     case 6:
      //Enter an expense charged by an employee
      // Storing Values in variables
      System.out.println("Enter Employee SSN:");
      emp_ssn = Integer.parseInt(br.readLine());
      System.out.println("Enter Expense Date - Please enter date in the form of 01-JAN-
2017:");
      String expense_date = br.readLine();
      System.out.println("Enter Expense Amount:");
      int expense_amnt = Integer.parseInt(br.readLine());
      System.out.println("Enter Expense Description:");
      String expense_descr = br.readLine();
      query = "Insert into emp_expense values('" + emp_ssn + "','" + expense_date + "','" +
expense_amnt + "','" + expense_descr + "')";
      st.executeUpdate(query);
      System.out.println("Database records for Expense charged by an employee emp_expense
table has been inserted");
      break;

     case 7:
      //Enter a new organization and associate it to one or more PAN teams
      // Storing Values in variables
      System.out.println("Enter Organization Name:");
      String org_name = br.readLine();
      System.out.println("Enter Mail Address:");
      String mail_addr = br.readLine();
      System.out.println("Enter Phone Number:");
      Long org_phn_num = Long.parseLong(br.readLine());
      System.out.println("Enter Contact Person:");
      String contact_person = br.readLine();
      System.out.println("Enter Organization Anonymus Status:");
      String org_anon_status = br.readLine();
      query = "Insert into organizations values('" + org_name + "','" + mail_addr + "','" +
org_phn_num + "','" + contact_person + "','" + org_anon_status + "')";
      st.executeUpdate(query);
      System.out.println("How many teams you want to associate\t" + org_name +
"\tOrganization?");
      num = Integer.parseInt(br.readLine());
      for (int i = 1; i <= num; i++) {
       System.out.println("Enter Team Name:");
       team_name = br.readLine();
       query1 = "insert into sponsors values('" + team_name + "','" + org_name + "')";
       st.executeUpdate(query1);
      }
      System.out.println("Database records for Organizations table has been inserted and\t"
+ org_name + "\tassociated with one or more teams");
      break;

     case 8:
      //Enter a new donor and associate him or her with several donations
      // Storing Values in variables
      System.out.println("Enter Donor SSN:");
      int donor_ssn = Integer.parseInt(br.readLine());
      System.out.println("Enter Donor Anonymus Status:");
      String donor_anon_status = br.readLine();
      query = "Insert into donor values('" + donor_ssn + "','" + donor_anon_status + "')";
      st.executeUpdate(query);
      System.out.println("How many donations you want to associate\t" + donor_ssn +
"\tdonor?");
      num = Integer.parseInt(br.readLine());
      for (int i = 1; i <= num; i++) {
       System.out.println("Enter Donate Date - Please enter date in the form of 01-JAN-
2017:");
       String donate_date = br.readLine();
       System.out.println("Enter Donate Amount:");
       int donate_amount = Integer.parseInt(br.readLine());
       System.out.println("Enter Campaign Name:");
       String campain_name = br.readLine();
       query1 = "insert into donor_donations values('" + donor_ssn + "','" + donate_date +
"','" + donate_amount + "','" + campain_name + "')";
       st.executeUpdate(query1);
      }
      System.out.println("Database records for donor table has been inserted and\t" +
donor_ssn + "\tassociated with several donations");
      break;

     case 9:
      //Enter a new organization and associate it with several donations
      // Storing Values in variables
      System.out.println("Enter Organization Name:");
      org_name = br.readLine();
      System.out.println("Enter Mail Address:");
      mail_addr = br.readLine();
      System.out.println("Enter Organization Phone Number:");
      org_phn_num = Long.parseLong(br.readLine());
      System.out.println("Enter Contact Person:");
      contact_person = br.readLine();
      System.out.println("Enter Organization Anonymus Status:");
      org_anon_status = br.readLine();
      query = "Insert into organizations values('" + org_name + "','" + mail_addr + "','" +
org_phn_num + "','" + contact_person + "','" + org_anon_status + "')";
      st.executeUpdate(query);
      System.out.println("How many donations you want to associate\t" + org_name +
"\torganization?");
      num = Integer.parseInt(br.readLine());
      for (int i = 1; i <= num; i++) {
       System.out.println("Enter Donate Date - Please enter date in the form of 01-JAN-
2017:");
       String donate_date = br.readLine();
       System.out.println("Enter Donate Amount:");
       int donate_amount = Integer.parseInt(br.readLine());
       System.out.println("Enter Campaign Name:");
       String campain_name = br.readLine();
       query1 = "insert into org_donations values('" + org_name + "','" + donate_date +
"','" + donate_amount + "','" + campain_name + "')";
       st.executeUpdate(query1);
      }
      System.out.println("Database records for organizations table has been inserted and" +
org_name + "associated with several donations");
      break;

     case 10:
      //Retrieve the name and phone number of the doctor of a particular client
      System.out.println("Enter Client SSN");
      client_ssn = Integer.parseInt(br.readLine());
      query = "Select CLIENT_DOC_NAME, CLIENT_DOC_PHN from clients Where CLIENT_SSN = " +
client_ssn + "";
      ResultSet rs = st.executeQuery(query);
      System.out.println("CLIENT_DOC_NAME\tCLIENT_DOC_PHONE");
      System.out.println("--------------------------------");
      while (rs.next()) {
       System.out.println(rs.getString(1) + "\t|\t " + rs.getString(2));
      }
      break;

     case 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
      System.out.println("Employee Expense Charged Start Date - Please enter date in the
form of 01-JAN-2017:");
      String start_date = br.readLine();
      System.out.println("Employee Expense Charged End Date - Please enter date in the form
of 01-JAN-2017:");
      String end_date = br.readLine();
      query = "SELECT EMP_SSN,SUM(EXPENSE_AMNT) AS TOTAL_AMOUNT_EXPENSES FROM EMP_EXPENSE
WHERE EXPENSE_DATE BETWEEN '" + start_date + "' AND '" + end_date + "' GROUP BY EMP_SSN
ORDER BY TOTAL_AMOUNT_EXPENSES DESC";
      rs = st.executeQuery(query);
      System.out.println("EMP_SSN\tTOTAL_AMOUNT_EXPENSES");
      System.out.println("--------------------------------");
      while (rs.next()) {
       System.out.println(rs.getString(1) + "\t|\t " + rs.getString(2));
      }
      break;

     case 12:
      //Retrieve the list of volunteers that are members of teams that support a particular
client
      System.out.println("Enter Client SSN:");
      client_ssn = Integer.parseInt(br.readLine());
      query = "Select v.VOL_SSN from VOLUNTEER v\n" +
       "JOIN VOL_SERVE vs ON v.VOL_SSN = vs.VOL_SSN\n" +
       "JOIN TEAM t ON vs.TEAM_NAME = t.TEAM_NAME\n" +
       "JOIN TEAM_CARES tc ON t.TEAM_NAME = tc.TEAM_NAME\n" +
       "JOIN CLIENTS c  ON tc.CLIENT_SSN = c.CLIENT_SSN\n" +
       "WHERE c.CLIENT_SSN = " + client_ssn + "";
      rs = st.executeQuery(query);
      System.out.println("Volunteer SSN");
      System.out.println("----------------");
      while (rs.next()) {
       System.out.println(rs.getString(1));
      }
      break;

     case 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
      query = "SELECT NAME,MAIL_ADDR,EMAIL_ID,HOME_PHN,WORK_PHN,CELL_PHN FROM PERSON P\n" +
        "JOIN CLIENTS C ON P.SSN = C.CLIENT_SSN\n" +
        "JOIN TEAM_CARES TC ON C.CLIENT_SSN = TC.CLIENT_SSN\n" +
        "JOIN TEAM T ON TC.TEAM_NAME = T.TEAM_NAME\n" +
        "WHERE TC.TEAM_NAME IN \n" +
        "(SELECT TEAM_NAME FROM SPONSORS WHERE ORG_NAME BETWEEN 'B%' AND 'K%')";
      rs = st.executeQuery(query);
      System.out.println("Name \t Mail Address \t Email Id \t Home Phone \t Work Phone \t
Cell Phone");
      System.out.println("------------------------------------------------------------------
---------");
      while (rs.next()) {
       System.out.println(rs.getString(1) + "|\t " + rs.getString(2) + "|\t " +
rs.getString(3) + "|\t " + rs.getString(4) + "|\t " + rs.getString(5) + "|\t " +
rs.getString(6));
      }
      break;

     case 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
      query = "SELECT P.NAME, SUM(DD.DONATE_AMOUNT), D.DONOR_ANON_STATUS AS
SUM_OF_DONATE_AMOUNT FROM DONOR D \n" +
       "JOIN EMPLOYEE E ON D.DONOR_SSN = E.EMP_SSN\n" +
       "JOIN PERSON P ON P.SSN = E.EMP_SSN\n" +
       "JOIN DONOR_DONATIONS DD ON D.DONOR_SSN = DD.DONOR_SSN\n" +
       "GROUP BY P.NAME, D.DONOR_ANON_STATUS";
      rs = st.executeQuery(query);
      System.out.println("Name \t Sum of Donation \t Anonymus Status ");
      System.out.println("------------------------------------------");
      while (rs.next()) {
       System.out.println(rs.getString(1) + " |\t " + rs.getString(2) + " |\t " +
rs.getString(3));
      }
      break;

     case 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
      query = "SELECT
P.NAME,P.MAIL_ADDR,P.EMAIL_ID,P.HOME_PHN,P.WORK_PHN,P.CELL_PHN,VS.TEAM_NAME FROM PERSON
P\n" +
       "JOIN VOLUNTEER V ON P.SSN = V.VOL_SSN\n" +
       "JOIN VOL_SERVE VS ON V.VOL_SSN = VS.VOL_SSN\n" +
       "AND  VS.VOL_SSN IN \n" +
       "(SELECT VOL_SSN AS NO_OF_HOURS FROM VOL_SERVE  WHERE SERVE_MONTHS BETWEEN 3 AND 6
GROUP BY VOL_SSN)";
      rs = st.executeQuery(query);
      System.out.println("Name \t Mail Address \t Email Id \t Home Phone \t Work Phone \t
Cell Phone\t Team Name");
      System.out.println("------------------------------------------------------------------
---------");
      while (rs.next()) {
       System.out.println(rs.getString(1) + " |\t " + rs.getString(2) + " |\t " +
rs.getString(3) + " |\t " + rs.getString(4) + " | " + rs.getString(5) + " | " +
rs.getString(6) + " | " + rs.getString(7));
      }
      break;

     case 16:
      //Increase the salary by 10% of all employees to whom more than one team must report
      System.out.println("Executing Query 16.... ");
      query = "UPDATE EMPLOYEE SET EMP_SAL = EMP_SAL*1.1 WHERE EMP_SSN IN\n" +
       "(SELECT RE.EMP_SSN FROM EMPLOYEE E \n" +
       "JOIN REPORT_EMP RE ON E.EMP_SSN = RE.EMP_SSN\n" +
       "GROUP BY RE.EMP_SSN HAVING COUNT(RE.EMP_SSN)>= 2)";
      rs = st.executeQuery(query);
      System.out.println("Eligible Employees salary has been increased");
      query1 = "select * from EMPLOYEE";
      rs = st.executeQuery(query1);
      System.out.println("Emp SSN \t Emp Salary \t Marrital Status \t Hire Date ");
      System.out.println("---------------------------------------------------");
      while (rs.next()) {
       System.out.println(rs.getString(1) + " |\t " + rs.getString(2) + " |\t " +
rs.getString(3) + " |\t " + rs.getString(4));
      }
      break;

     case 17:
      //Delete all clients who do not have health insurance and whose value of importance
for transportation is less than 5
      System.out.println("Executing query 17.... ");
      query = "DELETE FROM CLIENTS WHERE CLIENT_SSN IN\n" +
       "(SELECT C.CLIENT_SSN FROM CLIENTS C\n" +
       "JOIN COVERED_BY CB ON CB.CLIENT_SSN = C.CLIENT_SSN\n" +
       "JOIN INSURANCE_POLICY IP ON IP.POLICY_ID =CB.POLICY_ID AND IP.POLICY_TYPE !=
'HEALTH' \n" +
       "INTERSECT \n" +
       "SELECT C.CLIENT_SSN FROM CLIENTS C\n" +
       "JOIN CLIENT_NEED CN ON CN.CLIENT_SSN = C.CLIENT_SSN AND  CN.NEED_TYPE
='TRANSPORTATION'\n" +
       "WHERE CN.IMPORTANCE < 5)";
      rs = st.executeQuery(query);
      System.out.println("Clients who do not have health insurance and whose transport
importance less than 5 records deleted");
      break;

     case 18:
      /* Import: Enter new team from a data file */
      System.out.println("Enter import file name: ");
      String file_name = br.readLine();
      FileInputStream fstream = new
      FileInputStream("/Users/sumithkumargannarapu/Desktop/" + file_name);
      DataInputStream in = new DataInputStream(fstream);
      BufferedReader br1 = new BufferedReader(new InputStreamReader( in ));
      String strLine;
      while ((strLine = br1.readLine()) != null) {
       String a[] = strLine.split("\\t+");
       team_name = a[0];
       team_type = a[1];
       String team_date = a[2];
       st.executeUpdate("insert into team values('" + team_name + "','" + team_type + "','"
+ team_date + "')");
       System.out.println(" One row inserted succesfully");
      }
      System.out.println("File imported succesfully!!!");
      break;
     case 19:
      /* Export: Retrieve name and mail address and output them to a data file */
      ResultSet rs2 = st.executeQuery("select name,mail_addr from person");
      String string = null;
      System.out.println("Enter output file name: ");
      file_name = br.readLine();
      BufferedWriter
      export = new BufferedWriter(new FileWriter("/Users/sumithkumargannarapu/Desktop/" +
file_name));
      while (rs2.next()) {
       string = rs2.getString("name") + " " + rs2.getString("mail_addr");
       export.write("\n");
       export.write(string);
       export.write("\n");
       System.out.println(" One row inserted succesfully!");
      }
      System.out.println(" File exported succesfully!!!");
      export.close();
      break;
     case 20:
      //Close the statement
      st.close();
      //close the database connection
      conn.close();
      //Terminate the program
      System.exit(0);
     default:
      //default message indicates user to give a chance to enter between 1 to 20
      System.out.println("Select Options between 1-20");
    }
   }
  } catch (Exception e) {
   e.printStackTrace();
  }
 }
}
```