INDIVIDUAL PROJECT: A PATIENT ASSISTANT NETWORK DATABASE SYSTEM

Course Name: Database Management Systems
Course Number: CS4513
Semester: Fall
Year: 2020
Instructor: Dr. Le Gruenwald

Author: Shehnaz Begum Shaik
ID: 113476910
Email: shehnazshaik@ou.edu

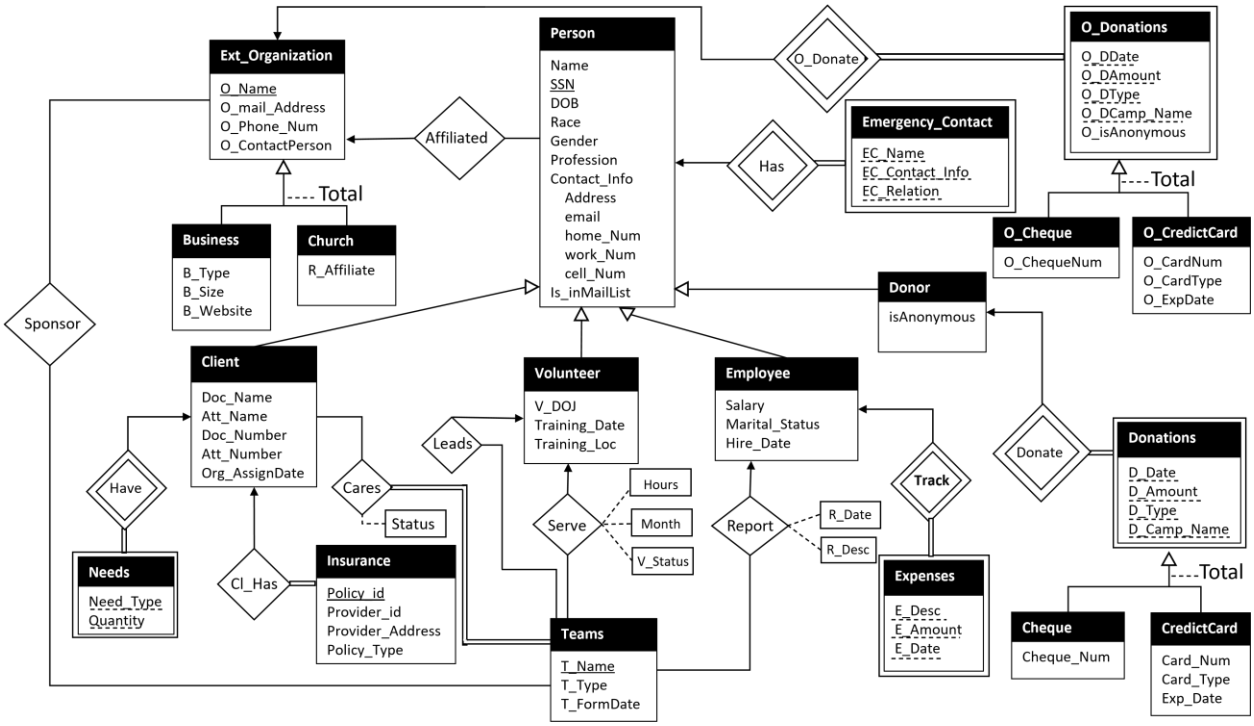
Table of Contents

shehnazshaik@ou.edu

shehnazshaik@ou.edu

Task 1.

1.1 ER Diagram



1.2 Relational Database Schema

Person(SSN, Name, DOB, Race, Gender, Profession, Address, email, home_Num, work_Num, cell_Num, is_inMailList)

Client(SSN, Doc_Name, Att_Name, Doc_Number, Att_Number, Org_AssignDate)

Volunteer(SSN, V_DOJ, Training_Date, Traing_Loc)

Employee(SSN, Salary, Marital_Status, Hire_Date)

Donor(SSN, isAnonymous)

Ext_Organization(O_Name, O_mail_Address, O_Phone_Num, O_ContactPerson)

Business(O_Name, B_Type, B_Size, B_Website)

Church(O_Name, R_Affiliate)

Affilated(SSN, O_Name)

Needs(SSN, Need_Type, Quantity)

Insurance(Policy_id, Provider_id, Provider_Address, Policy_Type, SSN)

Teams(T_Name, T_Type, T_FormDate)

shehnazshaik@ou.edu

Sponsor(O_Name, T_Name)

Serve(SSN, T_Name, Hours, Month, V_Status)

Cares(SSN, T_Name, Status)

Leads(SSN, T_Name)

Report(SSN, T_Name, R_Date, R_Desc)

Expenses(SSN, E_Desc, E_Amount, E_Date)

Emergency_Contact(SSN, EC_Name, EC_Contact_Info, EC_Relation)

O_Cheque(O_Name, O_DDate, O_DAmount, O_DType, O_DCamp_Name, O_isAnonymous, O_ChequeNum)

O_CreditCard(O_Name, O_DDate, O_DAmount, O_DType, O_DCamp_Name, O_isAnonymous, O_CardNum, O_CardType, O_ExpDate)

Cheque(SSN, D_Date, D_Amount, D_Type, D_Camp_Name, ChequeNum)

CreditCard(SSN, D_Date, D_Amount, D_Type, D_Camp_Name, Card_Num, Card_Type, Exp_Type)

Task 2. Data Dictionary

Affiliated

Documentation	cs-dsa-4513-sql-db@shai0002.database.windows.net
Schema	dbo
Name	Affiliated
Type	Table

Columns

	Name	Key	Data type	Null	Attributes	References	Description
1	SSN	🔑	int			Person	
2	O_Name	🔑	varchar(20)			Ext_Organization	

Relations

Foreign table		Primary table	Join	Title / Name / Description
Affiliated	>—	Ext_Organization	**Affiliated**.O_Name = Ext_Organization.O_Name	FK__Affiliate__O_Nam__5772F790
Affiliated	>—	Person	**Affiliated**.SSN = Person.SSN	FK__Affiliated__SSN__567ED357

Unique keys

	Key name	Columns	Description
🔑	PK__Affiliat__A494632F874B0D96	SSN, O_Name	

shehnazshaik@ou.edu

Business

Documentation cs-dsa-4513-sql-db@shai0002.database.windows.net
Schema dbo
Name Business
Type Table

Columns

	Name	Key	Data type	Null	Attributes	References	Description
1	O_Name	🔑	varchar(20)			Ext_Organization	
2	B_Type		varchar(20)				
3	B_Size		int				
4	B_Website		varchar(64)				

Relations

Foreign table		Primary table	Join	Title / Name / Description
Business	>—	Ext_Organization	**Business**.O_Name = Ext_Organization.O_Name	FK__Business__O_Name__50C5FA01

Unique keys

	Key name	Columns	Description
🔑	PK__Business__E8AED12725AE4E1B	O_Name	

Cares

Documentation cs-dsa-4513-sql-db@shai0002.database.windows.net
Schema dbo
Name Cares
Type Table

Columns

	Name	Key	Data type	Null	Attributes	References	Description
1	SSN	🔑	int			Client	
2	T_Name	🔑	varchar(20)			Teams	
3	Status		varchar(10)				

Relations

Foreign table		Primary table	Join	Title / Name / Description
Cares	>—	Client	**Cares**.SSN = Client.SSN	FK__Cares__SSN__6A85CC04
Cares	>—	Teams	**Cares**.T_Name = Teams.T_Name	FK__Cares__T_Name__6B79F03D

Unique keys

	Key name	Columns	Description
🔑	PK__Cares__B62477F6815315DE	SSN, T_Name	

shehnazshaik@ou.edu

Cheque

Documentation cs-dsa-4513-sql-db@shai0002.database.windows.net
Schema dbo
Name Cheque
Type Table

Columns

	Name	Key	Data type	Null	Attributes	References	Description
1	SSN	🔑	int			Donor	
2	D_Date	🔑	date				
3	D_Amount	🔑	real				
4	D_Type	🔑	varchar(20)				
5	D_Camp_Name	🔑	varchar(15)				
6	ChequeNum		int				

Relations

Foreign table		Primary table	Join	Title / Name / Description
Cheque	>—	Donor	**Cheque**.SSN = Donor.SSN	FK__Cheque__SSN__0169315C

Unique keys

	Key name	Columns	Description
🔑	PK__Cheque__0FF0B2FB2A4144A7	SSN, D_Date, D_Amount, D_Type, D_Camp_Name	

Church

Documentation cs-dsa-4513-sql-db@shai0002.database.windows.net
Schema dbo
Name Church
Type Table

Columns

	Name	Key	Data type	Null	Attributes	References	Description
1	O_Name	🔑	varchar(20)			Ext_Organization	
2	R_Affiliate		varchar(20)				

Relations

Foreign table		Primary table	Join	Title / Name / Description
Church	>—	Ext_Organization	**Church**.O_Name = Ext_Organization.O_Name	FK__Church__O_Name__53A266AC

Unique keys

	Key name	Columns	Description
🔑	PK__Church__E8AED12799E3B1DD	O_Name	

shehnazshaik@ou.edu

Client

Documentation cs-dsa-4513-sql-db@shai0002.database.windows.net
Schema dbo
Name Client
Type Table

Columns

	Name	Key	Data type	Null	Attributes	References	Description
1	SSN	🔑	int			Person	
2	Doc_Name		varchar(32)				
3	Att_Name		varchar(32)				
4	Doc_Number		char(10)	✔			
5	Att_Number		char(10)	✔			
6	Org_AssignDate		date				

Relations

Foreign table	Primary table	Join	Title / Name / Description
Client	Person	**Client**.SSN = Person.SSN	FK__Client__SSN__436BFEE3
Cares	Client	Cares.SSN = **Client**.SSN	FK__Cares__SSN__6A85CC04
Insurance	Client	Insurance.SSN = **Client**.SSN	FK__Insurance__SSN__5D2BD0E6
Needs	Client	Needs.SSN = **Client**.SSN	FK__Needs__SSN__5A4F643B

Unique keys

	Key name	Columns	Description
🔑	PK__Client__CA1E8E3DBB4FB987	SSN	

CreditCard

Documentation cs-dsa-4513-sql-db@shai0002.database.windows.net
Schema dbo
Name CreditCard
Type Table

Columns

	Name	Key	Data type	Null	Attributes	References	Description
1	SSN	🔑	int			Donor	
2	D_Date	🔑	date				
3	D_Amount	🔑	real				
4	D_Type	🔑	varchar(20)				
5	D_Camp_Name	🔑	varchar(15)				
6	Card_Num		char(10)				
7	Card_Type		varchar(20)				
8	Exp_Type		date				

Relations

Foreign table	Primary table	Join	Title / Name / Description
CreditCard	Donor	**CreditCard**.SSN = Donor.SSN	FK__CreditCard__SSN__04459E07

Unique keys

	Key name	Columns	Description
🔑	PK__CreditCa__0FF0B2FB84F86616	SSN, D_Date, D_Amount, D_Type, D_Camp_Name	

shehnazshaik@ou.edu

Donor

Documentation cs-dsa-4513-sql-db@shai0002.database.windows.net
Schema dbo
Name Donor
Type Table

Columns

	Name	Key	Data type	Null	Attributes	References	Description
1	SSN	🔑	int			Person	
2	isAnonymous		bit	✔			

Relations

Foreign table		Primary table	Join	Title / Name / Description
Donor	>—	Person	**Donor**.SSN = Person.SSN	FK__Donor__SSN__4C0144E4
Cheque	>—	Donor	Cheque.SSN = **Donor**.SSN	FK__Cheque__SSN__0169315C
CreditCard	>—	Donor	CreditCard.SSN = **Donor**.SSN	FK__CreditCard__SSN__04459E07

Unique keys

	Key name	Columns	Description
🔑	PK__Donor__CA1E8E3DFB15428D	SSN	

Emergency_Contact

Documentation cs-dsa-4513-sql-db@shai0002.database.windows.net
Schema dbo
Name Emergency_Contact
Type Table

Columns

	Name	Key	Data type	Null	Attributes	References	Description
1	SSN	🔑	int			Person	
2	EC_Name	🔑	varchar(20)				
3	EC_Contact_Info	🔑	char(10)				
4	EC_Relation	🔑	varchar(15)				

Relations

Foreign table		Primary table	Join	Title / Name / Description
Emergency_Contact	>—	Person	**Emergency_Contact**.SSN = Person.SSN	FK__Emergency_C__SSN__78D3EB5B

Unique keys

	Key name	Columns	Description
🔑	PK__Emergenc__FAC210D20A0388CF	SSN, EC_Name, EC_Contact_Info, EC_Relation	

shehnazshaik@ou.edu

Employee

Documentation cs-dsa-4513-sql-db@shai0002.database.windows.net
Schema dbo
Name Employee
Type Table

Columns

	Name	Key	Data type	Null	Attributes	References	Description
1	SSN	🔑	int			Person	
2	Salary		real				
3	Marital_Status		varchar(32)				
4	Hire_Date		date				

Relations

Foreign table		Primary table	Join	Title / Name / Description
Employee	>—	Person	**Employee**.SSN = Person.SSN	FK__Employee__SSN__4924D839
Expenses	>—	Employee	Expenses.SSN = **Employee**.SSN	FK__Expenses__SSN__75F77EB0
Report	>—	Employee	Report.SSN = **Employee**.SSN	FK__Report__SSN__7226EDCC

Unique keys

	Key name	Columns	Description
🔑	PK__Employee__CA1E8E3DA70D7C40	SSN	

Expenses

Documentation cs-dsa-4513-sql-db@shai0002.database.windows.net
Schema dbo
Name Expenses
Type Table

Columns

	Name	Key	Data type	Null	Attributes	References	Description
1	SSN	🔑	int			Employee	
2	E_Desc	🔑	varchar(20)				
3	E_Amount	🔑	real				
4	E_Date	🔑	date				

Relations

Foreign table		Primary table	Join	Title / Name / Description
Expenses	>—	Employee	**Expenses**.SSN = Employee.SSN	FK__Expenses__SSN__75F77EB0

Unique keys

	Key name	Columns	Description
🔑	PK__Expenses__E53336349005BCD3	SSN, E_Desc, E_Amount, E_Date	

shehnazshaik@ou.edu

Ext_Organization

Documentation cs-dsa-4513-sql-db@shai0002.database.windows.net
Schema dbo
Name Ext_Organization
Type Table

Columns

	Name	Key	Data type	Null	Attributes	References	Description
1	O_Name	🔑	varchar(20)				
2	O_mail_Address		varchar(20)				
3	O_Phone_Num		char(10)				
4	O_ContactPerson		varchar(32)				

Relations

Foreign table		Primary table	Join	Title / Name / Description
Affiliated	>—	Ext_Organization	Affiliated.O_Name = **Ext_Organization**.O_Name	FK__Affiliate__O_Nam__5772F790
Business	>—	Ext_Organization	Business.O_Name = **Ext_Organization**.O_Name	FK__Business__O_Name__50C5FA01
Church	>—	Ext_Organization	Church.O_Name = **Ext_Organization**.O_Name	FK__Church__O_Name__53A266AC
O_Cheque	>—	Ext_Organization	O_Cheque.O_Name = **Ext_Organization**.O_Name	FK__O_Cheque__O_Name__7BB05806
O_CreditCard	>—	Ext_Organization	O_CreditCard.O_Name = **Ext_Organization**.O_Name	FK__O_CreditC__O_Nam__7E8CC4B1
Sponsor	>—	Ext_Organization	Sponsor.O_Name = **Ext_Organization**.O_Name	FK__Sponsor__O_Name__62E4AA3C

Unique keys

	Key name	Columns	Description
🔑	PK__Ext_Orga__E8AED127D36FFE74	O_Name	

Insurance

Documentation cs-dsa-4513-sql-db@shai0002.database.windows.net
Schema dbo
Name Insurance
Type Table

Columns

	Name	Key	Data type	Null	Attributes	References	Description
1	Policy_id	🔑	int				
2	Provider_id		int				
3	Provider_Address		varchar(32)				
4	Policy_Type		varchar(20)				
5	SSN		int	✔		Client	

Relations

Foreign table		Primary table	Join	Title / Name / Description
Insurance	>—	Client	**Insurance**.SSN = Client.SSN	FK__Insurance__SSN__5D2BD0E6

Unique keys

	Key name	Columns	Description
🔑	PK__Insuranc__4569BF19861BA308	Policy_id	

shehnazshaik@ou.edu

Leads

Documentation cs-dsa-4513-sql-db@shai0002.database.windows.net
Schema dbo
Name Leads
Type Table

Columns

	Name	Key	Data type	Null	Attributes	References	Description
1	SSN	🔑	int			Volunteer	
2	T_Name		varchar(20)			Teams	

Relations

Foreign table		Primary table	Join	Title / Name / Description
Leads	>—	Teams	**Leads**.T_Name = Teams.T_Name	FK__Leads__T_Name__6F4A8121
Leads	>—	Volunteer	**Leads**.SSN = Volunteer.SSN	FK__Leads__SSN__6E565CE8

Unique keys

	Key name	Columns	Description
🔑	PK__Leads__CA1E8E3DF7FA56C6	SSN	

Needs

Documentation cs-dsa-4513-sql-db@shai0002.database.windows.net
Schema dbo
Name Needs
Type Table

Columns

	Name	Key	Data type	Null	Attributes	References	Description
1	SSN	🔑	int			Client	
2	Need_Type	🔑	varchar(32)				
3	Quantity	🔑	int				

Relations

Foreign table		Primary table	Join	Title / Name / Description
Needs	>—	Client	**Needs**.SSN = Client.SSN	FK__Needs__SSN__5A4F643B

Unique keys

	Key name	Columns	Description
🔑	PK__Needs__0BCCBFA3382DA604	SSN, Need_Type, Quantity	

shehnazshaik@ou.edu

O_Cheque

Documentation cs-dsa-4513-sql-db@shai0002.database.windows.net
Schema dbo
Name O_Cheque
Type Table

Columns

	Name	Key	Data type	Null	Attributes	References	Description
1	O_Name	🔑	varchar(20)			Ext_Organization	
2	O_DDate	🔑	date				
3	O_DAmount	🔑	real				
4	O_DType	🔑	varchar(20)				
5	O_DCamp_Name	🔑	varchar(15)				
6	O_isAnonymous		bit				
7	O_ChequeNum		int				

Relations

Foreign table		Primary table	Join	Title / Name / Description
O_Cheque	>—	Ext_Organization	**O_Cheque**.O_Name = Ext_Organization.O_Name	FK__O_Cheque__O_Name__7BB05806

Unique keys

	Key name	Columns	Description
🔑	PK__O_Cheque__A044A2CC7CAD9DF8	O_Name, O_DDate, O_DAmount, O_DType, O_DCamp_Name	

O_CreditCard

Documentation cs-dsa-4513-sql-db@shai0002.database.windows.net
Schema dbo
Name O_CreditCard
Type Table

Columns

	Name	Key	Data type	Null	Attributes	References	Description
1	O_Name	🔑	varchar(20)			Ext_Organization	
2	O_DDate	🔑	date				
3	O_DAmount	🔑	real				
4	O_DType	🔑	varchar(20)				
5	O_DCamp_Name	🔑	varchar(15)				
6	O_isAnonymous		bit				
7	O_CardNum		char(10)				
8	O_CardType		varchar(20)				
9	O_ExpDate		date				

Relations

Foreign table		Primary table	Join	Title / Name / Description
O_CreditCard	>—	Ext_Organization	**O_CreditCard**.O_Name = Ext_Organization.O_Name	FK__O_CreditC__O_Nam__7E8CC4B1

Unique keys

	Key name	Columns	Description
🔑	PK__O_Credit__A044A2CC52424BE1	O_Name, O_DDate, O_DAmount, O_DType, O_DCamp_Name	

shehnazshaik@ou.edu

Person

Documentation cs-dsa-4513-sql-db@shai0002.database.windows.net
Schema dbo
Name Person
Type Table

Columns

	Name	Key	Data type	Null	Attributes	References	Description
1	SSN	🔑	int				
2	Name		varchar(32)				
3	DOB		date				
4	Race		varchar(32)				
5	Gender		varchar(32)				
6	Profession		varchar(32)				
7	Address		varchar(64)				
8	email		varchar(32)				
9	home_Num		char(10)	✓			
10	work_Num		char(10)	✓			
11	cell_Num		char(10)	✓			
12	is_inMailList		bit	✓			

Relations

Foreign table	Primary table	Join	Title / Name / Description
Affiliated	Person	Affiliated.SSN = **Person**.SSN	FK__Affiliated__SSN__567ED357
Client	Person	Client.SSN = **Person**.SSN	FK__Client__SSN__436BFEE3
Donor	Person	Donor.SSN = **Person**.SSN	FK__Donor__SSN__4C0144E4
Emergency_Contact	Person	Emergency_Contact.SSN = **Person**.SSN	FK__Emergency_C__SSN__78D3EB5B
Employee	Person	Employee.SSN = **Person**.SSN	FK__Employee__SSN__4924D839
Volunteer	Person	Volunteer.SSN = **Person**.SSN	FK__Volunteer__SSN__46486B8E

Unique keys

	Key name	Columns	Description
🔑	PK__Person__CA1E8E3D7A24D185	SSN	

shehnazshaik@ou.edu

Report

Documentation cs-dsa-4513-sql-db@shai0002.database.windows.net
Schema dbo
Name Report
Type Table

Columns

	Name	Key	Data type	Null	Attributes	References	Description
1	SSN	🔑	int			Employee	
2	T_Name	🔑	varchar(20)			Teams	
3	R_Date		date				
4	R_Desc		varchar(32)				

Relations

Foreign table		Primary table	Join	Title / Name / Description
Report	>—	Employee	**Report**.SSN = Employee.SSN	FK__Report__SSN__7226EDCC
Report	>—	Teams	**Report**.T_Name = Teams.T_Name	FK__Report__T_Name__731B1205

Unique keys

	Key name	Columns	Description
🔑	PK__Report__B62477F6E6DD9C77	SSN, T_Name	

Serve

Documentation cs-dsa-4513-sql-db@shai0002.database.windows.net
Schema dbo
Name Serve
Type Table

Columns

	Name	Key	Data type	Null	Attributes	References	Description
1	SSN	🔑	int			Volunteer	
2	T_Name	🔑	varchar(20)			Teams	
3	Hours		int				
4	Month		varchar(15)				
5	V_Status		varchar(10)				

Relations

Foreign table		Primary table	Join	Title / Name / Description
Serve	>—	Teams	**Serve**.T_Name = Teams.T_Name	FK__Serve__T_Name__67A95F59
Serve	>—	Volunteer	**Serve**.SSN = Volunteer.SSN	FK__Serve__SSN__66B53B20

Unique keys

	Key name	Columns	Description
🔑	PK__Serve__B62477F66442B200	SSN, T_Name	

shehnazshaik@ou.edu

Sponsor

Documentation	cs-dsa-4513-sql-db@shai0002.database.windows.net
Schema	dbo
Name	Sponsor
Type	Table

Columns

	Name	Key	Data type	Null	Attributes	References	Description
1	O_Name	🔑	varchar(20)			Ext_Organization	
2	T_Name	🔑	varchar(20)			Teams	

Relations

Foreign table		Primary table	Join	Title / Name / Description
Sponsor	>—	Ext_Organization	**Sponsor**.O_Name = Ext_Organization.O_Name	FK__Sponsor__O_Name__62E4AA3C
Sponsor	>—	Teams	**Sponsor**.T_Name = Teams.T_Name	FK__Sponsor__T_Name__63D8CE75

Unique keys

	Key name	Columns	Description
🔑	PK__Sponsor__949428ECCE8EA2B0	O_Name, T_Name	

Teams

Documentation	cs-dsa-4513-sql-db@shai0002.database.windows.net
Schema	dbo
Name	Teams
Type	Table

Columns

	Name	Key	Data type	Null	Attributes	References	Description
1	T_Name	🔑	varchar(20)				
2	T_Type		varchar(32)				
3	T_FormDate		date				

Relations

Foreign table		Primary table	Join	Title / Name / Description
Cares	>—	Teams	Cares.T_Name = **Teams**.T_Name	FK__Cares__T_Name__6B79F03D
Leads	>—	Teams	Leads.T_Name = **Teams**.T_Name	FK__Leads__T_Name__6F4A8121
Report	>—	Teams	Report.T_Name = **Teams**.T_Name	FK__Report__T_Name__731B1205
Serve	>—	Teams	Serve.T_Name = **Teams**.T_Name	FK__Serve__T_Name__67A95F59
Sponsor	>—	Teams	Sponsor.T_Name = **Teams**.T_Name	FK__Sponsor__T_Name__63D8CE75

Unique keys

	Key name	Columns	Description
🔑	PK__Teams__C3AF9CB6CD41A544	T_Name	

shehnazshaik@ou.edu

Volunteer

Documentation	cs-dsa-4513-sql-db@shai0002.database.windows.net
Schema	dbo
Name	Volunteer
Type	Table

Columns

	Name	Key	Data type	Null	Attributes	References	Description
1	SSN	🔑	int			Person	
2	V_DOJ		date				
3	Training_Date		date				
4	Traing_Loc		varchar(20)				

Relations

Foreign table		Primary table	Join	Title / Name / Description
Volunteer	>—	Person	**Volunteer**.SSN = Person.SSN	FK__Volunteer__SSN__46486B8E
Leads	>—	Volunteer	Leads.SSN = **Volunteer**.SSN	FK__Leads__SSN__6E565CE8
Serve	>—	Volunteer	Serve.SSN = **Volunteer**.SSN	FK__Serve__SSN__66B53B20

Unique keys

	Key name	Columns	Description
🔑	PK__Voluntee__CA1E8E3D216BF304	SSN	

Task 3.

3.1 Discussion of storage structures for tables

Table Name	Type of Query	Search Key	Frequency	File Organization(FO)	Justification
Person	13. Random Search 14. Random Search	SSN SSN	1/week 1/week	Dynamic Hashing with hash key **SSN**	Since there are more random search queries with good frequency.
Client	2. Insert 10. Random Search 17. Random Search Delete	SSN SSN SSN SSN	1/week 1/week 4/year	Dynamic Hashing with hash key **SSN**	Since there are insertions and random search queries.
Volunteer	3. Insert		2/month	Heap File	Less Frequent queries.
Employee	5. Insert 14. Random Search 16. Update	SSN SSN	1/year 1/week 1/year	Indexing Sequential file organization on search key **SSN**	Less Frequent queries with random search on fair frequency.

shehnazshaik@ou.edu

Donor	8. Insert 14. Random Search	SSN	1/day 1/week	Dynamic Hashing with hash key **SSN**	Since there are very frequent insertions and random search queries.
Affiliated				Heap File	Since There are no active query calls.
Ext_Organization	7. Insert 9. Insert		2/week 1/day	Heap File	Since there are only insertion queries.
Needs	17. Random Search Random Search Random Search	SSN Need_Type Quantity	4/year	Dynamic Hashing with hash key **SSN**	Less frequent queries but, More random search so Hashing yields better performance.
Insurance	17. Random Search Delete	Policy_Type SSN	4/year	Dynamic Hashing with hash key **SSN**	Less frequent queries but, More random search so Hashing yields better performance.
Teams	1. Insert 15. Range Search	T_FormDate	1/week 1/month	Index Sequential file.	Less frequent with insert and range, index sequential FO is ideal.
Sponsor	7. Insert 13. Range Search	O_Name	2/week 1/week	B+ Tree File Organization and **T_Name** as search key (or) Indexing on search key **T_Name**	Frequent query with inser and Range search, B+ FO yield better performance.
Serve	3. Insert 4. Insert 12. Random Search	T_Name	2/month 30/month 4/year	Dynamic Hashing with hash key **T_Name**	Less frequent query with more insertion and random search, index sequential file FO will be ideal.

shehnazshaik@ou.edu

Cares	2. Insert 12. Random Search 13. Random Search	SSN T_Name 	1/week 4/year 1/week	Dynamic Hashing with hash key **SSN**	More frequent insert and random search but 1 less frequent query. Hashing is ideal.
Leads				Heap File	Since There are no active query calls.
Business				Heap File	Since There are no active query calls.
Church				Heap File	Since There are no active query calls.
Report	5. Insert 16. Random Search	SSN	1/year 1/year	Index Sequential file.	Less frequent queries but, for insert and random search this FO is ideal.
Expenses	6. Insert 11. Range Search	E_Date	1/day 1/month	B+ Tree File Organization and E_Date as search key (or) Indexing on search key **E_Date**	Frequent query with inser and Range search, B+ FO yield better performance.
Emergency_Contact				Heap File	Since There are no active query calls.
O_Cheque	9. Insert		1/day	Heap File	Since there are only insertion queries.
O_CreditCard	9. Insert		1/day	Heap File	Since there are only insertion queries.
Cheque	8. Insert		1/day	Heap File	Since there are only insertion queries.
CreditCard	8. Insert		1/day	Heap File	Since there are only insertion queries.

3.2 Discussion of choices of storage structures for tables

shehnazshaik@ou.edu

Table Name	Type of Query	Search Key	Frequency	Decision
Person	13. Random Search 14. Random Search	SSN SSN	1/week 1/week	No indexing since, Azure will create index on primary key **SSN**
Client	2. Insert 10. Random Search 17. Random Search Delete	SSN SSN SSN SSN	1/week 1/week 4/year	No need of indexing
Volunteer	3. Insert		2/month	No need of indexing
Employee	5. Insert 14. Random Search 16. Update	SSN SSN	1/year 1/week 1/year	No need of indexing
Donor	8. Insert 14. Random Search	SSN	1/day 1/week	No need of indexing
Affiliated				No need of indexing
Ext_Organization	7. Insert 9. Insert		2/week 1/day	No need of indexing
Needs	17. Random Search Random Search Random Search	SSN Need_Type Quantity	4/year	Indexing on SSN, Need_Type, Quantity
Insurance	17. Random Search Delete	Policy_Type SSN	4/year	Indexing on Policy_Type
Teams	1. Insert 15. Range Search	T_FormDate	1/week 1/month	No need of indexing
Sponsor	7. Insert 13. Range Search	O_Name	2/week 1/week	Indexing on O_Name

shehnazshaik@ou.edu

Serve	3. Insert 4. Insert 12. Random Search	T_Name	2/month 30/month 4/year	Indexing on T_Name
Cares	2. Insert 12. Random Search 13. Random Search	SSN T_Name	1/week 4/year 1/week	Indexing on T_Name
Leads				No need of indexing
Business				No need of indexing
Church				No need of indexing
Report	5. Insert 16. Random Search	SSN	1/year 1/year	No need of indexing
Expenses	6. Insert 11. Range Search	E_Date	1/day 1/month	Indexing on E_Date
Emergency_Contact				No need of indexing
O_Cheque	9. Insert		1/day	No need of indexing
O_CreditCard	9. Insert		1/day	No need of indexing
Cheque	8. Insert		1/day	No need of indexing
CreditCard	8. Insert		1/day	No need of indexing

Task 4. SQL Statements and Screenshots

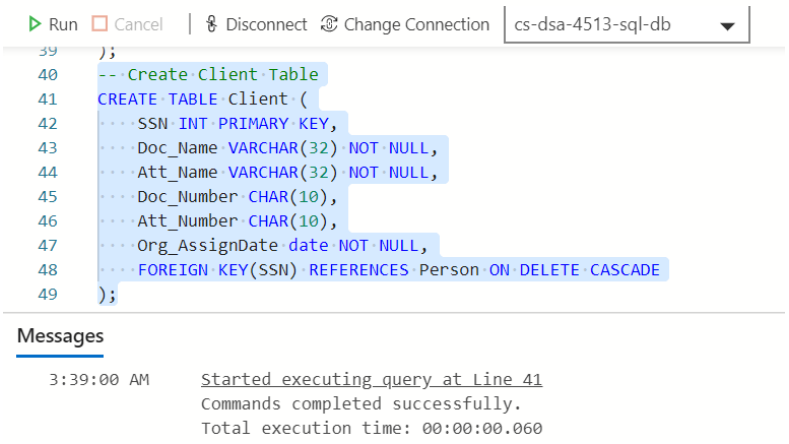
SQL Statements and screenshots of All Tables and Indexes.

```
-- Create Person Table
CREATE TABLE Person (
    SSN INT PRIMARY KEY,
    Name VARCHAR(32) NOT NULL,
    DOB date NOT NULL,
    Race VARCHAR(32) NOT NULL,
    Gender VARCHAR(32) NOT NULL,
```


shehnazshaik@ou.edu

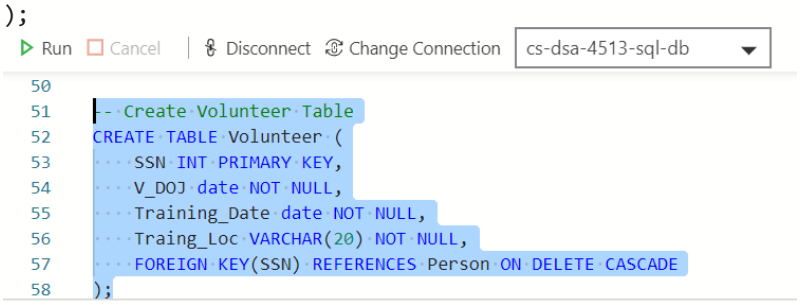
```
    Profession  VARCHAR(32) NOT NULL,
    Address  VARCHAR(64) NOT NULL,
    email VARCHAR(32) NOT NULL,
    home_Num CHAR(10),
    work_Num CHAR(10),
    cell_Num CHAR(10),
    is_inMailList BIT
);
```


```
-- Create Client Table
CREATE TABLE Client (
    SSN INT PRIMARY KEY,
    Doc_Name VARCHAR(32) NOT NULL,
    Att_Name VARCHAR(32) NOT NULL,
    Doc_Number CHAR(10),
    Att_Number CHAR(10),
    Org_AssignDate date NOT NULL,
    FOREIGN KEY(SSN) REFERENCES Person ON DELETE CASCADE
);
```



shehnazshaik@ou.edu

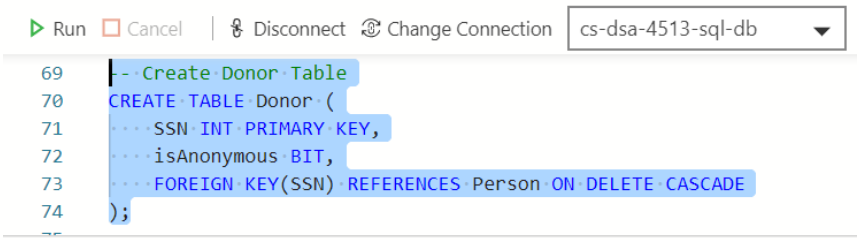
```sql
-- Create Volunteer Table
CREATE TABLE Volunteer (
    SSN INT PRIMARY KEY,
    V_DOJ date NOT NULL,
    Training_Date date NOT NULL,
    Traing_Loc VARCHAR(20) NOT NULL,
    FOREIGN KEY(SSN) REFERENCES Person ON DELETE CASCADE
);
```



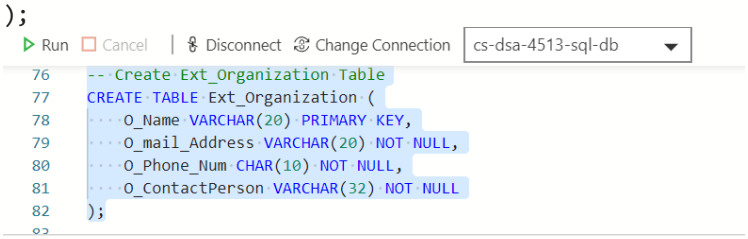

```sql
-- Create Employee Table
CREATE TABLE Employee (
    SSN INT PRIMARY KEY,
    Salary REAL NOT NULL,
    Marital_Status VARCHAR(32) NOT NULL,
    Hire_Date date NOT NULL,
    FOREIGN KEY(SSN) REFERENCES Person ON DELETE CASCADE
);
```


```sql
-- Create Donor Table
CREATE TABLE Donor (
    SSN INT PRIMARY KEY,
    isAnonymous BIT,
    FOREIGN KEY(SSN) REFERENCES Person ON DELETE CASCADE
);
```

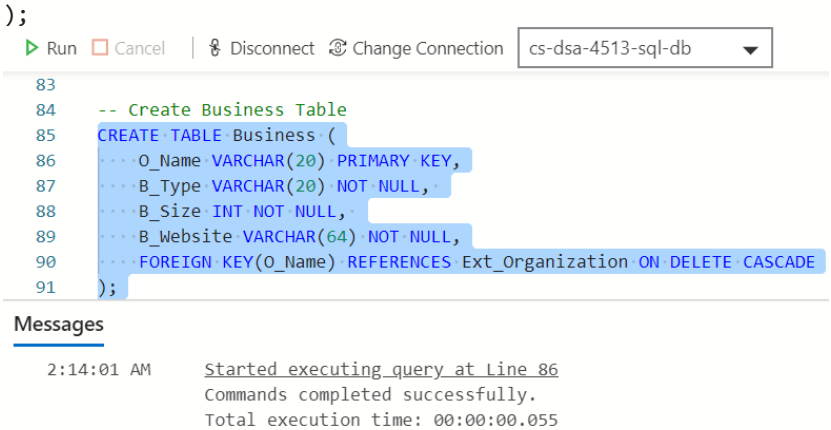

shehnazshaik@ou.edu




```sql
-- Create Ext_Organization Table
CREATE TABLE Ext_Organization (
    O_Name VARCHAR(20) PRIMARY KEY,
    O_mail_Address VARCHAR(20) NOT NULL,
    O_Phone_Num CHAR(10) NOT NULL,
    O_ContactPerson VARCHAR(32) NOT NULL
);
```



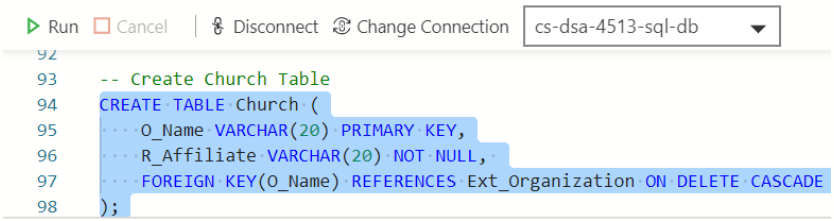

```sql
-- Create Business Table
CREATE TABLE Business (
    O_Name VARCHAR(20) PRIMARY KEY,
    B_Type VARCHAR(20) NOT NULL,
    B_Size INT NOT NULL,
    B_Website VARCHAR(64) NOT NULL,
    FOREIGN KEY(O_Name) REFERENCES Ext_Organization ON DELETE CASCADE
);
```



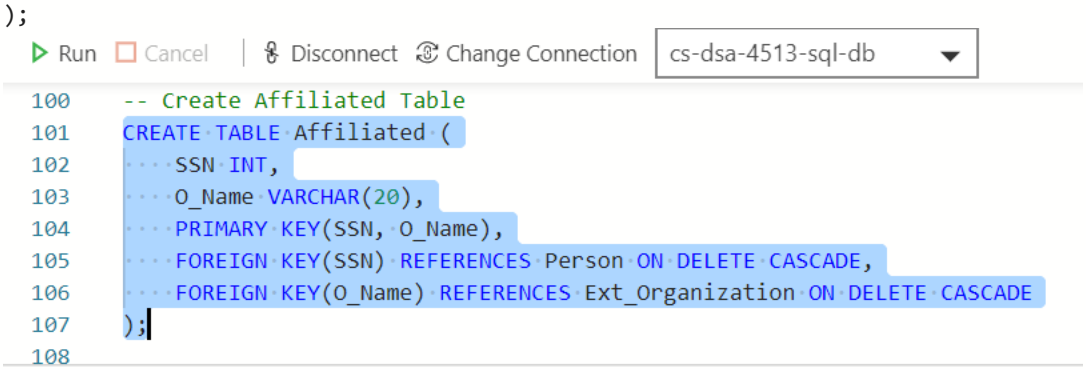

```sql
-- Create Church Table
```


shehnazshaik@ou.edu

```
CREATE TABLE Church (
    O_Name VARCHAR(20) PRIMARY KEY,
    R_Affiliate VARCHAR(20) NOT NULL,
    FOREIGN KEY(O_Name) REFERENCES Ext_Organization ON DELETE CASCADE
);
```

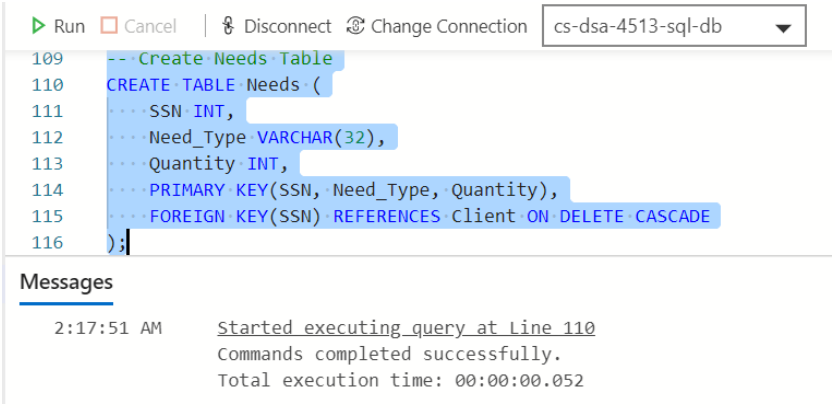



```
-- Create Affiliated Table
CREATE TABLE Affiliated (
    SSN INT,
    O_Name VARCHAR(20),
    PRIMARY KEY(SSN, O_Name),
    FOREIGN KEY(SSN) REFERENCES Person ON DELETE CASCADE,
    FOREIGN KEY(O_Name) REFERENCES Ext_Organization ON DELETE CASCADE
);
```

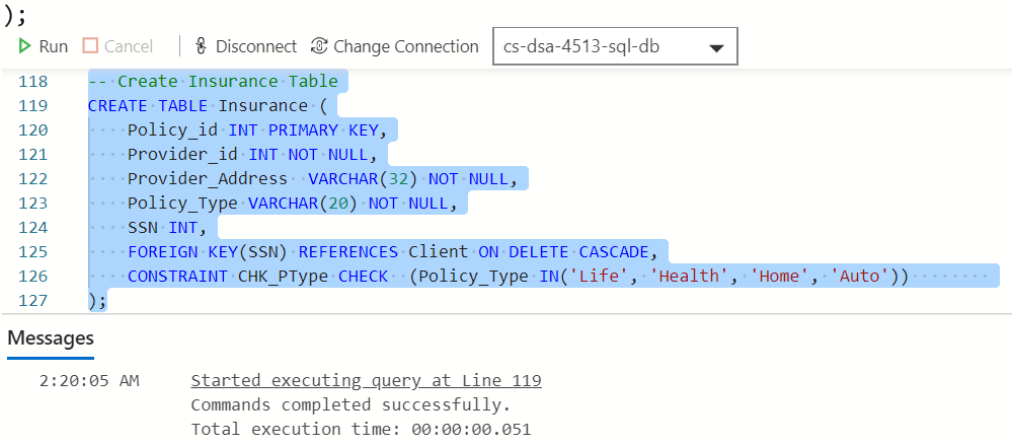



```
-- Create Needs Table
CREATE TABLE Needs (
    SSN INT,
    Need_Type VARCHAR(32),
    Quantity INT,
    PRIMARY KEY(SSN, Need_Type, Quantity),
    FOREIGN KEY(SSN) REFERENCES Client ON DELETE CASCADE
);
```


shehnazshaik@ou.edu

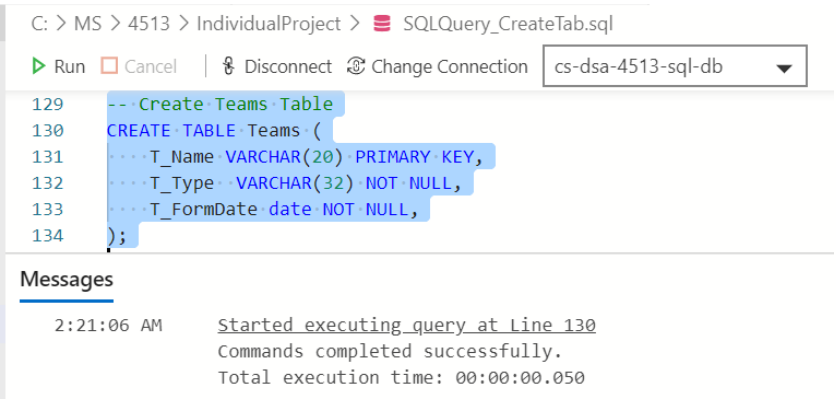



```
-- Create Insurance Table
CREATE TABLE Insurance (
    Policy_id INT PRIMARY KEY,
    Provider_id INT NOT NULL,
    Provider_Address  VARCHAR(32) NOT NULL,
    Policy_Type VARCHAR(20) NOT NULL,
    SSN INT,
    FOREIGN KEY(SSN) REFERENCES Client ON DELETE CASCADE
    CONSTRAINT CHK_PType CHECK  (Policy_Type IN('Life', 'Health', 'Home', 'Auto'))
);
```

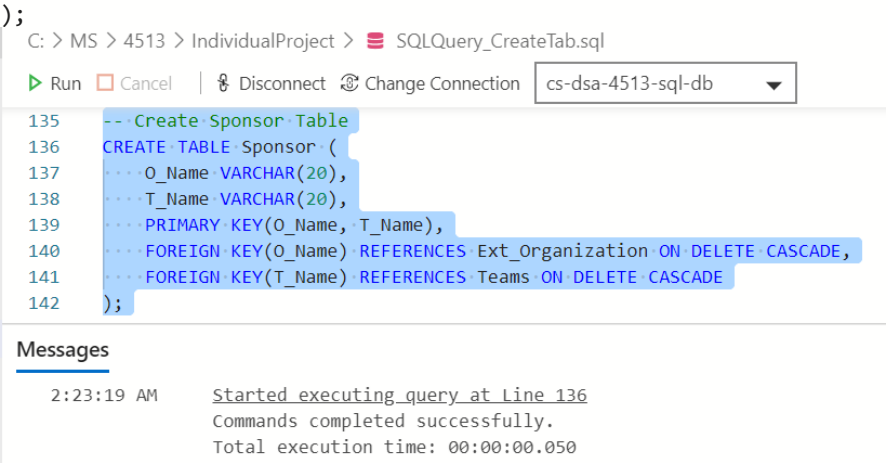



```
-- Create Teams Table
CREATE TABLE Teams (
    T_Name VARCHAR(20) PRIMARY KEY,
    T_Type  VARCHAR(32) NOT NULL,
    T_FormDate date NOT NULL,
);
```


shehnazshaik@ou.edu



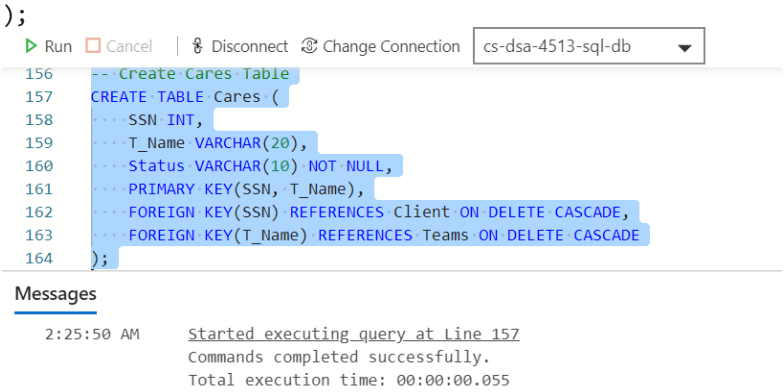

```sql
-- Create Sponsor Table
CREATE TABLE Sponsor (
    O_Name VARCHAR(20),
    T_Name VARCHAR(20),
    PRIMARY KEY(O_Name, T_Name),
    FOREIGN KEY(O_Name) REFERENCES Ext_Organization ON DELETE CASCADE,
    FOREIGN KEY(T_Name) REFERENCES Teams ON DELETE CASCADE
);
```




```sql
-- Create Serve Table
CREATE TABLE Serve (
    SSN INT,
    T_Name VARCHAR(20),
    Hours INT NOT NULL,
    Month VARCHAR(15) NOT NULL,
    V_Status VARCHAR(10) NOT NULL,
    PRIMARY KEY(SSN, T_Name),
    FOREIGN KEY(SSN) REFERENCES Volunteer ON DELETE CASCADE,
    FOREIGN KEY(T_Name) REFERENCES Teams ON DELETE CASCADE
);
```

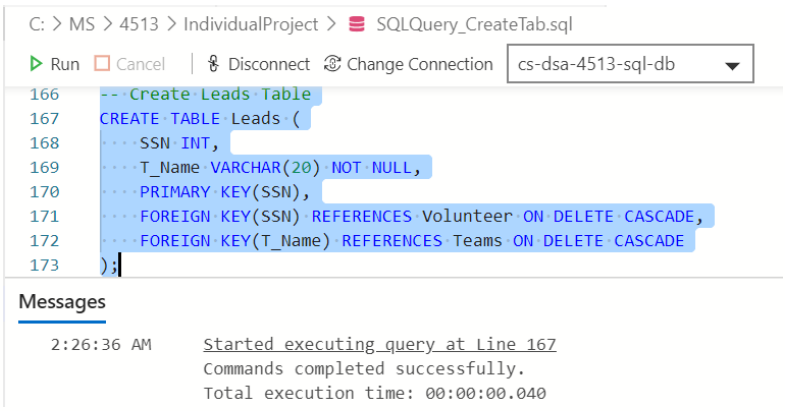

shehnazshaik@ou.edu


```sql
-- Create Cares Table
CREATE TABLE Cares (
    SSN INT,
    T_Name VARCHAR(20),
    Status VARCHAR(10) NOT NULL,
    PRIMARY KEY(SSN, T_Name),
    FOREIGN KEY(SSN) REFERENCES Client ON DELETE CASCADE,
    FOREIGN KEY(T_Name) REFERENCES Teams ON DELETE CASCADE
);
```

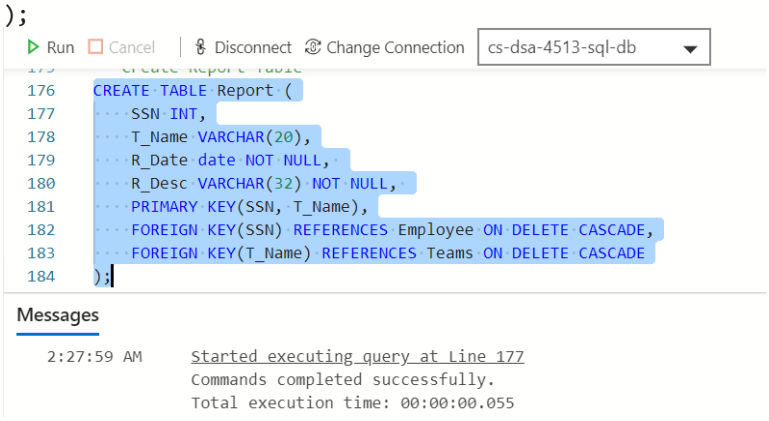



```sql
-- Create Leads Table
CREATE TABLE Leads (
    SSN INT,
    T_Name VARCHAR(20) NOT NULL,
    PRIMARY KEY(SSN),
    FOREIGN KEY(SSN) REFERENCES Volunteer ON DELETE CASCADE,
    FOREIGN KEY(T_Name) REFERENCES Teams ON DELETE CASCADE
);
```


shehnazshaik@ou.edu

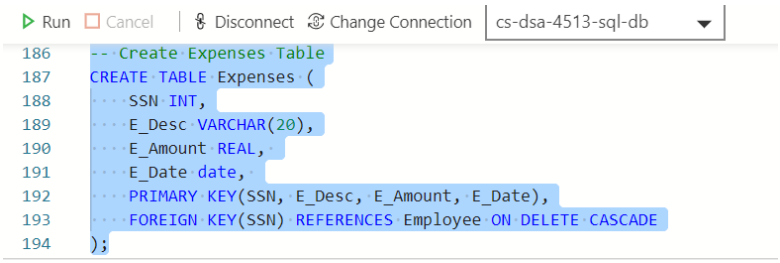



```sql
-- Create Report Table
CREATE TABLE Report (
    SSN INT,
    T_Name VARCHAR(20),
    R_Date date NOT NULL,
    R_Desc VARCHAR(32) NOT NULL,
    PRIMARY KEY(SSN, T_Name),
    FOREIGN KEY(SSN) REFERENCES Employee ON DELETE CASCADE,
    FOREIGN KEY(T_Name) REFERENCES Teams ON DELETE CASCADE
);
```

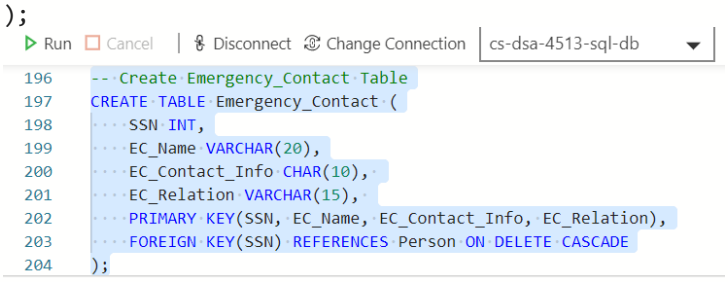



```sql
-- Create Expenses Table
CREATE TABLE Expenses (
    SSN INT,
    E_Desc VARCHAR(20),
    E_Amount REAL,
    E_Date date,
    PRIMARY KEY(SSN, E_Desc, E_Amount, E_Date),
    FOREIGN KEY(SSN) REFERENCES Employee ON DELETE CASCADE
);
```


shehnazshaik@ou.edu

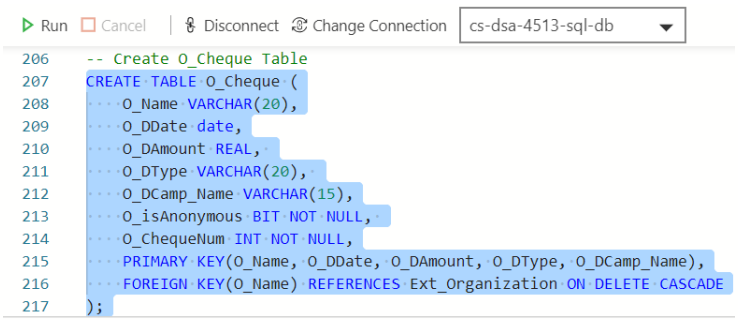



```sql
-- Create Emergency_Contact Table
CREATE TABLE Emergency_Contact (
    SSN INT,
    EC_Name VARCHAR(20),
    EC_Contact_Info CHAR(10),
    EC_Relation VARCHAR(15),
    PRIMARY KEY(SSN, EC_Name, EC_Contact_Info, EC_Relation),
    FOREIGN KEY(SSN) REFERENCES Person ON DELETE CASCADE
);
```

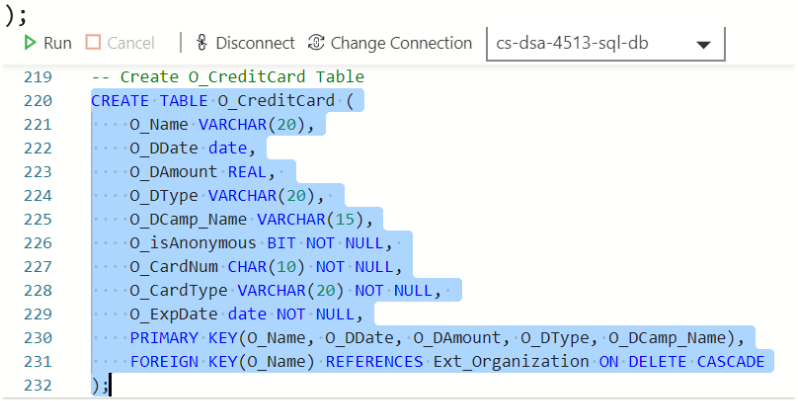



```sql
-- Create O_Cheque Table
CREATE TABLE O_Cheque (
    O_Name VARCHAR(20),
    O_DDate date,
    O_DAmount REAL,
    O_DType VARCHAR(20),
    O_DCamp_Name VARCHAR(15),
    O_isAnonymous BIT NOT NULL,
    O_ChequeNum INT NOT NULL,
    PRIMARY KEY(O_Name, O_DDate, O_DAmount, O_DType, O_DCamp_Name),
    FOREIGN KEY(O_Name) REFERENCES Ext_Organization ON DELETE CASCADE
);
```


shehnazshaik@ou.edu



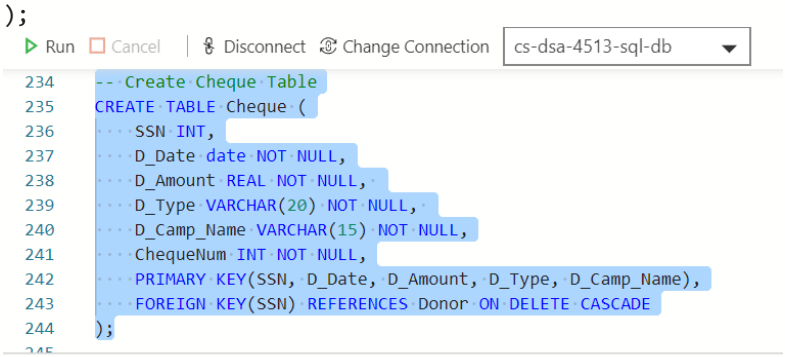

```sql
-- Create O_CreditCard Table
CREATE TABLE O_CreditCard (
    O_Name VARCHAR(20),
    O_DDate date,
    O_DAmount REAL,
    O_DType VARCHAR(20),
    O_DCamp_Name VARCHAR(15),
    O_isAnonymous BIT NOT NULL,
    O_CardNum CHAR(10) NOT NULL,
    O_CardType VARCHAR(20) NOT NULL,
    O_ExpDate date NOT NULL,
    PRIMARY KEY(O_Name, O_DDate, O_DAmount, O_DType, O_DCamp_Name),
    FOREIGN KEY(O_Name) REFERENCES Ext_Organization ON DELETE CASCADE
);
```



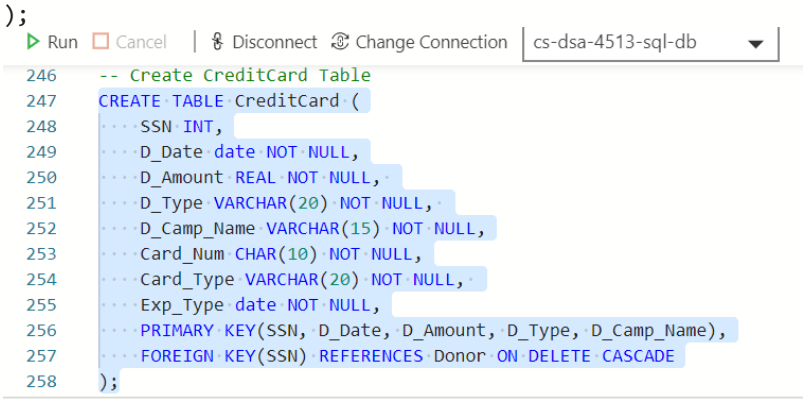

```sql
-- Create Cheque Table
CREATE TABLE Cheque (
    SSN INT,
    D_Date date NOT NULL,
    D_Amount REAL NOT NULL,
    D_Type VARCHAR(20) NOT NULL,
    D_Camp_Name VARCHAR(15) NOT NULL,
    ChequeNum INT NOT NULL,
    PRIMARY KEY(SSN, D_Date, D_Amount, D_Type, D_Camp_Name),
```


shehnazshaik@ou.edu

```
    FOREIGN KEY(SSN) REFERENCES Donor ON DELETE CASCADE
);
```




```
-- Create CreditCard Table
CREATE TABLE CreditCard (
    SSN INT,
    D_Date date NOT NULL,
    D_Amount REAL NOT NULL,
    D_Type VARCHAR(20) NOT NULL,
    D_Camp_Name VARCHAR(15) NOT NULL,
    Card_Num CHAR(10) NOT NULL,
    Card_Type VARCHAR(20) NOT NULL,
    Exp_Type date NOT NULL,
    PRIMARY KEY(SSN, D_Date, D_Amount, D_Type, D_Camp_Name),
    FOREIGN KEY(SSN) REFERENCES Donor ON DELETE CASCADE
);
```

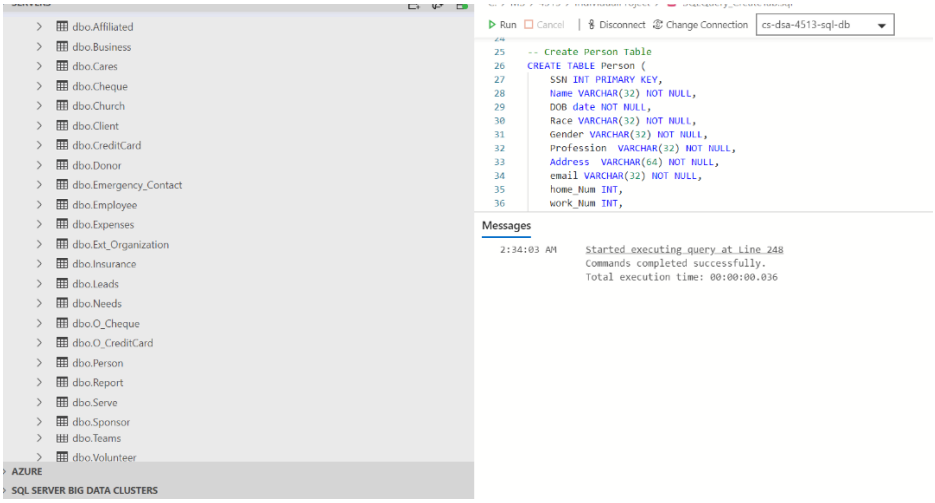



```
-- Creating Indices
CREATE index needs on Needs(Need_Type,Quantity);
CREATE index ipType on Insurance(Policy_type);
CREATE index oName on Sponsor(O_name);
CREATE index tName on Serve(T_name);
CREATE index care1 on Cares(SSN);
CREATE index care2 on Cares(T_name);
```


shehnazshaik@ou.edu

```
CREATE index eDate on Expense(E_date);
```

All Tables are successfully created.



Task 5.

5. 1 SQL Statements – Error Checking

Error Checking Sample Queries:

```
-- Q1
INSERT INTO Teams VALUES('RCB', 'IPL','01-Apr-2013');
--Q2
INSERT INTO Client VALUES(4513001, 'Karlo', 'Maya', 4051112221, 4051112221, '01-Feb-2017');
--Q3
INSERT INTO Volunteer VALUES(4513007, '12-Mar-2017', '17-Sep-2017', 'Dallas');
INSERT INTO Serve VALUES(4513007, 'CSK', 15, 'January', 'InActive');
--Q4
INSERT INTO Serve VALUES(4513007, 'OKC', 50, 'October','Active');
--Q5
INSERT INTO Employee VALUES (4513003, 67000, 'Married', '13-Mar-2018');
INSERT INTO Report VALUES (4513003, 'CSK', '31-May-2019', 'Passed');
--Q6
INSERT INTO Expenses VALUES (4513007, 'Food',  1000,'15-Jun-2019');
--Q7
INSERT INTO Ext_Organization VALUES ('BPF','San Diego',8095741110,'Chris');
INSERT INTO Sponsor VALUES ('BPF', 'OU');
--Q8
INSERT INTO Donor VALUES (4513001, 0);
INSERT INTO Cheque VALUES (4513001, '30-Jun-2019', 200,  'Savings', 'OCD4', 123404);
INSERT INTO CreditCard VALUES (4513001, '01-Jan-
2019', 300,  'Savings', 'OCD1',  9723899893,'Visa', '28-Feb-2021');
--Q9
INSERT INTO Ext_Organization VALUES ('Pepsico','New York',8095749000,'Indranooi');
INSERT INTO O_Cheque VALUES ('Pepsico', '31-Jan-2019', 3000,  'Crowd Fund', 'OC1', 0, 123001);
```


shehnazshaik@ou.edu

```
INSERT INTO O_CreditCard VALUES ('Philips', '31-Jan-
2019', 2000,  'Crowd Fund', 'OC1', 0, 9723899891,'Visa', '31-Jan-2021';
--Q10
Select Doc_Name, Doc_Number from Client Where SSN = 4513002;
--Q11
Select SSN, SUM(E_Amount) from Expenses
    Where E_Date BETWEEN '01-Jun-2019' AND '31-Dec-2019'
    GROUP BY SSN ORDER BY SUM(E_Amount);
--Q12
SELECT SSN, Name FROM Person
    WHERE SSN in (SELECT SSN FROM Serve WHERE T_name in
    (SELECT T_name FROM Cares WHERE SSN='4513005'));
--Q13
SELECT Name, Address, email, home_Num, work_Num, cell_Num FROM Person WHERE SSN in
    (SELECT SSN FROM Cares WHERE T_Name in
    (SELECT T_Name FROM Sponsor WHERE O_Name BETWEEN 'B%' AND 'K%'));
--Q14
SELECT P.Name, sum(DO.D_Amount) tot_Amt, D.isAnonymous as DO_Amt
                                FROM Donor D, Employee E, Person P,
                                (SELECT SSN, D_Amount from CreditCard
                                UNION ALL
                                SELECT SSN, D_Amount from Cheque) as DO
                                WHERE P.SSN   = D.SSN
                                AND E.SSN = D.SSN
                                AND DO.SSN = D.SSN
                                GROUP BY D.SSN, P.Name, D.isAnonymous
                                ORDER BY DO_Amt;
--Q15
Select T_Name from Teams WHERE T_FormDate > '01-Jan-2011';
--Q16
UPDATE Employee SET Salary = (1.1*Salary) WHERE SSN IN (SELECT SSN FROM Report GROUP BY SSN Ha
ving count(SSN) > 1);
--Q17
DELETE FROM Client WHERE SSN NOT IN
    (SELECT SSN FROM Insurance WHERE Policy_Type = 'Health')
    AND SSN IN
    (SELECT SSN FROM Needs WHERE Need_Type = 'Transportation' AND Quantity < 5)
```

5.2 Java Program

```
import java.sql.Connection;

import java.sql.Statement;

import java.util.Scanner;

import java.sql.CallableStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.DriverManager;
```


shehnazshaik@ou.edu

```
import java.sql.CallableStatement;

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.io.FileNotFoundException;

import java.io.IOException;


public class task5IP {

	public static void main(String[] args) throws SQLException {

		// Database credentials

		final String HOSTNAME = "shai0002.database.windows.net";

		final String DBNAME = "cs-dsa-4513-sql-db";

		final String USERNAME = "shai0002";

		final String PASSWORD = "$Shhane031";

		// Database connection string

		final String URL =
String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;hostNameIn
Certificate=*.database.windows.net;loginTimeout=30;", HOSTNAME, DBNAME, USERNAME,
PASSWORD);

		final String PROMPT = "\n WELCOME TO THE PATIENT ASSISTANT NETWORK DATABASE
SYSTEM \n"+

					"\nPlease select one of the options below: \n" +

					"1) Enter a new team into the database (1/month); \n" +

					"2) Enter a new client into the database and associate him or her with
one or more teams (1/week).; \n" +

					"3) Enter a new volunteer into the database and associate him or her
with one or more teams (2/month).; \n" +
```


shehnazshaik@ou.edu

```
"4) Enter the number of hours a volunteer worked this month for a particular team (30/month).; \n" +

"5) Enter a new employee into the database and associate him or her with one or more teams (1/year).; \n" +

"6) Enter an expense charged by an employee (1/day).; \n" +

"7) Enter a new organization and associate it to one or more PAN teams (2/week).; \n" +

"8) Enter a new donor and associate him or her with several donations (1/day).; \n" +

"9) Enter a new organization and associate it with several donations (1/day).; \n" +

"10) Retrieve the name and phone number of the doctor of a particular client (1/week).; \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 (1/month).; \n" +

"12) Retrieve the list of volunteers that are members of teams that support a particular client (4/year).; \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 (1/week).; \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 (1/week).; \n" +

"15) Retrieve the names of all teams that were founded after a particular date (1/month).; \n" +

"16) Increase the salary by 10% of all employees to whom more than one team must report. (1/year); \n" +

"17) Delete all clients who do not have health insurance and whose value of importance for transportation is less than 5 (4/year).; \n" +

"18) Import: enter new teams from a data file until the file is empty (the user must 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 must be asked to enter the output file name).; \n" +
```


shehnazshaik@ou.edu

```
                "20) Quit";

        try (final Connection connection = DriverManager.getConnection(URL)) {
            final String schema = connection.getSchema();
            System.out.println("Successfully connected to DB " + schema);
            System.out.println("-------------------------------------------------");
            System.out.println(PROMPT);
            int option = 0;
            Scanner in = new Scanner(System.in);
            //Choosing an option from 1 to 20
            while(option != 20) {
                //Reading the users input for option
                option = in.nextInt();
                //defining what happens for each options using switch case
                switch(option) {
                  //1) Enter a new team into the database
                  case 1:
                      System.out.println("Enter the Team Name: \n");
                      in.nextLine();
                      String t_Name = in.nextLine();
                      System.out.println("Enter the Team Type: \n");
                      String t_Type = in.next();
                      System.out.println("Enter the Team Form date (DD-Mon-YYYY):
\n");
                      String t_FormDate = in.next();
                      //Insertion query String
                      final String insertsql = "INSERT INTO Teams values('"+ t_Name
+"', '" + t_Type + "','" + t_FormDate +"')";
                      Statement conn = connection.createStatement();
```


shehnazshaik@ou.edu

```
                    conn.execute(insertsql);
                    System.out.println("1 record successfully inserted");
                    break;
                //2) Enter a new Client into the database and associate him or her with one or more teams
                case 2:
                    System.out.println("Enter values for Option 2 \n");
                    System.out.println("Enter SSN of new Client\n");
                    String c_ssn = in.next();
                    System.out.println("Enter the Doctor Name: \n");
                    String doc_Name = in.next();
                    System.out.println("Enter the Attorney Name: \n");
                    String att_Name = in.next();
                    System.out.println("Enter the Doctor Number: \n");
                    String doc_Num = in.next();
                    System.out.println("Enter the Attorney Number: \n");
                    String att_Num = in.next();
                    System.out.println("Enter the Organization assigned date (DD-Mon-YYYY): \n");
                    String org_AssignDate = in.next();
                    //Insertion query String for Client Table
                    final String insertsql1 = "INSERT INTO Client values('"+ c_ssn +"', '" + doc_Name + "','" + att_Name +"' , '" + doc_Num + "', '" + att_Num + "', '" + org_AssignDate + "')";
                    conn = connection.createStatement();
                    conn.execute(insertsql1); //Executing the query string
                    System.out.println("New SSN Added to Client \n");
                    System.out.println("Enter i : No of Teams, newly added client is associated with: \n");
                    int n = in.nextInt();
```


shehnazshaik@ou.edu

```
                    for(int i = 1; i <= n; i++) {// for loop to assign to one or more
Teams
                        System.out.println("Enter the Team Name: \n");
                        String t_Name1 = in.next();
                        System.out.println("Enter Team Care Status: \n");
                        String c_status = in.next();
                        final String insertquery1 = "INSERT INTO Cares VALUES("
+ c_ssn + ",'" + t_Name1 + "','" + c_status + "')";
                        conn = connection.createStatement();
                        conn.execute(insertquery1); //Executing the query
string
                    }
                    System.out.println("Query 2 suuccessfully executed");
                    break;
                //3) Enter a new volunteer into the database and associate him or her
with one or more teams
                case 3:
                    System.out.println("Enter values for Option 3 \n");
                    //Entering a new Volunteer Who will then be assigned to Teams
                    System.out.println("Enter SSN of new Volunteer");
                    String v_ssn = in.next();
                    System.out.println("Enter the Volunteer Joining date (DD-Mon-
YYYY):");
                    String v_doj = in.next();
                    System.out.println("Enter the Volunteer Training date (DD-Mon-
YYYY):");
                    String v_Trainingdt = in.next();
                    System.out.println("Enter the Volunteer Training Location:");
                    String v_TrainLoc = in.next();
                    final String insertsql2 = "INSERT INTO Volunteer VALUES('"+
v_ssn +"', '" + v_doj + "','" + v_Trainingdt +"' , '" + v_TrainLoc + "');";
```


shehnazshaik@ou.edu

```
                    conn = connection.createStatement();
                    conn.execute(insertsql2); //Executing the query string, Now The
volunteer is added
                    System.out.println("Enter i : No of Teams, newly added
Volunteer serves: \n");
                    n = in.nextInt();
                    for(int i = 1; i <= n; i++) {//To assign more than one teams.
                        System.out.println("Enter the Team Name: \n");
                        in.nextLine();
                        String st_Name = in.nextLine();
                        System.out.println("Enter the No. of Hours: \n");
                        int hours = in.nextInt();
                        System.out.println("Enter the Name of the Month: \n");
                        String month = in.next();
                        System.out.println("Enter volunteer Status: \n");
                        String v_status = in.next();
                        final String insertquery3 = "INSERT INTO Serve VALUES("
+ v_ssn + ",'" + st_Name + "','" + hours + "','" + month + "','" + v_status + "');";
                        conn = connection.createStatement();
                        conn.execute(insertquery3); //Executing the query
string
                    }
                    System.out.println("Query 3 suuccessfully executed");
                    break;
                //4) Enter the number of hours a volunteer worked this month for a
particular team
                case 4:
                    System.out.println("Enter SSN of a Volunteer\n");
                    String v_ssn1 = in.next();
                    System.out.println("Enter the Team Name: \n");
```


shehnazshaik@ou.edu

```
                String vt_Name = in.next();
                System.out.println("Enter the No. of Hours: \n");
                int hours = in.nextInt();
                System.out.println("Enter the Name of the Month: \n");
                String month = in.next();
                System.out.println("Enter volunteer Status: \n");
                String v_status = in.next();
                //Adding a new record to volunteer table
                final String insertquery4 = "INSERT INTO Serve VALUES(" +
v_ssn1 + ",'" + vt_Name + "','" + hours + "','" + month + "','" + v_status + "');";
                conn = connection.createStatement();
                conn.execute(insertquery4); //Executing the query string
                System.out.println("Query  suuccessfully executed");
                break;
            //5. Enter a new employee into the database and associate him or her
with one or more teams
            case 5:
                System.out.println("Enter values for Option 5 \n");
                System.out.println("Enter SSN of new Employee\n");
                String e_ssn = in.next();
                System.out.println("Enter the Salary: \n");
                String e_salary = in.next();
                System.out.println("Enter the Marital Status: \n");
                String marital_status = in.next();
                System.out.println("Enter the Employee Hire date (DD-Mon-
YYYY): \n");
                String e_hiredate = in.next();
                //Insertion query String for Employee Table
                final String insertsql4 = "INSERT INTO Employee values('"+ e_ssn
+"', '" + e_salary + "','" + marital_status +"' , '" + e_hiredate + "');";
```


shehnazshaik@ou.edu

```
                    conn = connection.createStatement(); //Executing the query
string, Employee created
                    conn.execute(insertsql4);
                    System.out.println("Enter i : No of Teams, newly added
Employee is Associated to: \n");
                    n = in.nextInt();
                    for(int i = 1; i <= n; i++) {
                        System.out.println("Enter the Team Name: \n");
                        in.nextLine();
                        t_Name = in.nextLine();
                        System.out.println("Enter the report date (DD-Mon-
YYYY): \n");
                        String r_date = in.next();
                        System.out.println("Enter report description: \n");
                        String r_desc = in.next();
                        final String insertquery5 = "INSERT INTO Report
VALUES(" + e_ssn + ",'" + t_Name + "','" + r_date + "','" + r_desc + "')";
                        conn = connection.createStatement();
                        conn.execute(insertquery5); //Executing the query
string, Teams assigned
                    }
                    System.out.println("Query 5 suuccessfully executed");
                    break;
                //6) Enter an expense charged by an employee
                case 6:
                    System.out.println("Enter SSN of an Employee\n");
                    e_ssn = in.next();
                    System.out.println("Enter the Expense description: \n");
                    in.nextLine();
                    String e_desc = in.nextLine();
```


shehnazshaik@ou.edu

```
                                        System.out.println("Enter the expense Amount: \n");
                                        int e_Amt = in.nextInt();
                                        System.out.println("Enter the expense date (DD-Mon-YYYY):: \n");
                                        String e_date = in.next();
                                        final String insertquery6 = "INSERT INTO Expenses VALUES(" + e_ssn + ",'" + e_desc + "','" + e_Amt + "','" + e_date + "')";
                                        conn = connection.createStatement();
                                        conn.execute(insertquery6); //Executing the query string, new expense recorded
                                        System.out.println("Query 6 suuccessfully executed");
                                        break;
                                    //7) Enter a new organization and associate it to one or more PAN teams
                                    case 7:
                                        System.out.println("Enter values for Option 7 \n");
                                        System.out.println("Enter Name of new Organization\n");
                                        String o_name = in.next();
                                        System.out.println("Enter the mail address: \n");
                                        in.nextLine();
                                        String o_mailAddress = in.nextLine();
                                        System.out.println("Enter the Phone Number: \n");
                                        String o_PhNo = in.next();
                                        System.out.println("Enter the Contact Person \n");
                                        String o_ContPerson = in.next();
                                        //Insertion query String for Ext_Organization Table
                                        final String insertsql7 = "INSERT INTO Ext_Organization values('"+ o_name +"', '" + o_mailAddress + "','" + o_PhNo +"' , '" + o_ContPerson + "')";
                                        conn = connection.createStatement();
                                        conn.execute(insertsql7); //Executing the query string
```


shehnazshaik@ou.edu

```
                    System.out.println("Enter i : No of Teams, newly added
Organization sponsors: \n");
                    n = in.nextInt();
                    for(int i = 1; i <= n; i++) {//To add one or more teams
                        System.out.println("Enter the Team Name: \n");
                        String t_Name1 = in.next();
                        final String insertquery7 = "INSERT INTO Sponsor
VALUES('" + o_name + "','" + t_Name1 + "')";
                        conn = connection.createStatement();
                        conn.execute(insertquery7); //Executing the query
string
                    }
                    System.out.println("Query 7 suuccessfully executed");
                    break;
                //8) Enter a new donor and associate him or her with several
donations
                case 8:
                    System.out.println("Enter values for Option 8 \n");
                    System.out.println("Enter the SSN of new Donor\n");
                    String d_ssn = in.next();
                    System.out.println("Enter  \"1\" if donor is Anonymous, \"0\"  if
not: \n");
                    int d_isAnonymous = in.nextInt();
                    //Insertion query String for Donor Table
                    final String insertsql8 = "INSERT INTO Donor values('"+ d_ssn
+"', '" + d_isAnonymous + "')";
                    conn = connection.createStatement();
                    conn.execute(insertsql8);
                    System.out.println("Enter i : No of Donations, newly added
Donor makes: \n");
                    n = in.nextInt();
```


shehnazshaik@ou.edu

```
                    //For loop will enable assigning a donor to several donations
                    for(int i = 1; i <= n; i++) {
                        System.out.println("Enter \"1\" for Cheque and \"2\"
for Creditcard donation \n");
                        int type = in.nextInt();
                        //Using if statement to give user the choice
                        //to assign the donor to Cheque type donation or Card
type donation
                        if(type == 1) {
                            //Entering data for CHeque type donation
                            System.out.println("Enter the date of donation
(DD-Mon-YYYY):: \n");
                            String d_ddate = in.next();
                            System.out.println("Enter the Donation
Amount: \n");
                            String d_Amt = in.next();
                            System.out.println("Enter the Donation Type:
\n");
                            String d_Type = in.next();
                            System.out.println("Enter the Donation
Campaign Name: \n");
                            String d_CName = in.next();
                            System.out.println("Enter the Cheque Number:
\n");
                            String d_chqNum = in.next();
                            final String insertquery8 = "INSERT INTO Cheque
values('"+ d_ssn +"', '" + d_ddate + "','" + d_Amt +"' , '" + d_Type + "', '" + d_CName + "', '" + d_chqNum +
"')";
                            conn = connection.createStatement();
                            conn.execute(insertquery8);//Executing the
query string
                        }
```


shehnazshaik@ou.edu

```
                    else {
                        //Entering Data for Card type donation
                        System.out.println("Enter the date of donation
(DD-Mon-YYYY):: \n");
                        String d_ddate = in.next();
                        System.out.println("Enter the Donation
Amount: \n");
                        String d_Amt = in.next();
                        System.out.println("Enter the Donation Type:
\n");
                        String d_Type = in.next();
                        System.out.println("Enter the Donation
Campaign Name: \n");
                        String d_CName = in.next();
                        System.out.println("Enter the Credit card
Number: \n");
                        String d_CCNum = in.next();
                        System.out.println("Enter the Card Type: \n");
                        String d_CCType = in.next();
                        System.out.println("Enter the Card Expiry date
(DD-Mon-YYYY):: \n");
                        String d_CCExp = in.next();
                        final String insertquery8 = "INSERT INTO
CreditCard values('"+ d_ssn +"', '" + d_ddate + "','" + d_Amt +"' , '" + d_Type + "', '" + d_CName + "', '" +
d_CCNum + "', '" + d_CCType + "', '" + d_CCExp + "')";
                        conn = connection.createStatement();
                        conn.execute(insertquery8);//Executing the
query string
                    }
                }
                System.out.println("Query 8 suuccessfully executed");
                break;
```


shehnazshaik@ou.edu

```
//9) Enter a new organization and associate it with several donations
case 9:
    System.out.println("Enter values for Option 9 \n");
    System.out.println("Enter Name of new Organization\n");
    String og_name = in.next();
    System.out.println("Enter the mail address: \n");
    in.nextLine();
    String og_mailAddress = in.nextLine();
    System.out.println("Enter the Phone Number: \n");
    String og_PhNo = in.next();
    System.out.println("Enter the Contact Person \n");
    String og_ContPerson = in.next();
    //Insertion query String for Ext_Organization Table
    final String insertsql9 = "INSERT INTO Ext_Organization
values('"+ og_name +"', '" + og_mailAddress + "','" + og_PhNo +"' , '" + og_ContPerson + "')";
    conn = connection.createStatement();
    conn.execute(insertsql9);//Executing the query string
    System.out.println("Organization inserted suuccessfully");
    System.out.println("Enter i : No of Donations, newly added
Organization makes: \n");
    n = in.nextInt();
    //For loop to assign to several donations
    for(int i = 1; i <= n; i++) {
        System.out.println("Enter \"1\" for Cheque and \"2\"
for Creditcard donation \n");
        int type = in.nextInt();
        //Using if statement to give user the choice
        //to assign the Org to Cheque type donation or Card
type donation
        if(type == 1) {
```


shehnazshaik@ou.edu

```
                                                    //Entering the data for cheque donation
                                                    System.out.println("Enter the date of donation
(DD-Mon-YYYY):: \n");
                                                    String o_ddate = in.next();
                                                    System.out.println("Enter the Donation
Amount: \n");
                                                    String o_Amt = in.next();
                                                    System.out.println("Enter the Donation Type:
\n");
                                                    String o_Type = in.next();
                                                    System.out.println("Enter the Donation
Campaign Name: \n");
                                                    String o_CName = in.next();
                                                    System.out.println("Enter  \"1\" if Org is
Anonymous, \"0\"  if not: \n");
                                                    int o_isAnonymous = in.nextInt();
                                                    System.out.println("Enter the Cheque Number:
\n");
                                                    String o_chqNum = in.next();
                                                    final String insertquery9 = "INSERT INTO
O_Cheque values('"+ og_name +"', '" + o_ddate + "','" + o_Amt +"' , '" + o_Type + "', '" + o_CName + "', '"
+ o_isAnonymous + "', '" + o_chqNum + "')";
                                                    conn =
connection.createStatement();//Executing the query string
                                                    conn.execute(insertquery9);
                                            }
                                            else {
                                                    //Entering the data for Card donation
                                                    System.out.println("Enter the date of donation
(DD-Mon-YYYY):: \n");
                                                    String o_ddate = in.next();
                                                    System.out.println("Enter the Donation
Amount: \n");
```


shehnazshaik@ou.edu

```
                                                String o_Amt = in.next();
                                                System.out.println("Enter the Donation Type:
\n");
                                                String o_Type = in.next();
                                                System.out.println("Enter the Donation
Campaign Name: \n");
                                                String o_CName = in.next();
                                                System.out.println("Enter  \"1\" if Org is
Anonymous, \"0\"  if not: \n");
                                                int o_isAnonymous = in.nextInt();
                                                System.out.println("Enter the Credit card
Number: \n");
                                                String o_CCNum = in.next();
                                                System.out.println("Enter the Card Type: \n");
                                                String o_CCType = in.next();
                                                System.out.println("Enter the Card Expiry date
(DD-Mon-YYYY):: \n");
                                                String o_CCExp = in.next();
                                                final String insertquery9 = "INSERT INTO
O_CreditCard values('"+ og_name +"', '" + o_ddate + "','" + o_Amt +"' , '" + o_Type + "', '" + o_CName +
"', '" + o_isAnonymous + "', '" + o_CCNum + "', '"+ o_CCType +"', '"+ o_CCExp +"')";;
                                                conn = connection.createStatement();
                                                conn.execute(insertquery9);//Executing the
query string
                                        }
                                }
                                System.out.println("Query 9 suuccessfully executed");
                                break;
                        //10) Retrieve the name and phone number of the doctor of a
particular client 4513002
                        case 10:
                                System.out.println("Enter SSN of a Client\n");
```


shehnazshaik@ou.edu

```
                    c_ssn = in.next();
                    System.out.println("Retrieving Name and Ph. Number of Doctor
for a particular team");
                    final String query10 = "Select Doc_Name, Doc_Number from
Client Where SSN = " + c_ssn + ";";
                    conn = connection.createStatement();
                    ResultSet res = conn.executeQuery(query10); //Executing and
Storing the result set
                    System.out.println("Doctor_Name | Doctor_Phone\t");
                    System.out.println("----------------------------");
                    while (res.next()) {
                    System.out.println(res.getString(1)+ "  | " + res.getString(2));
                    }
                    System.out.println("Query 10 suuccessfully executed");
                    break;
                //11) Retrieve the total amount of expenses charged by each
employee for a particular period of time.
                case 11:
                    System.out.println("Enter the Start Date (DD-Mon-YYYY):\n");
                    String e_Startdate = in.next();
                    System.out.println("Enter the End Date (DD-Mon-YYYY):\n");
                    String e_Enddate = in.next();
                    final String query11 = "SELECT SSN, SUM(E_Amount) FROM
Expenses Where E_Date BETWEEN '"+ e_Startdate +"'AND  '"+ e_Enddate + "' GROUP BY SSN ORDER BY
SUM(E_Amount);";
                    conn = connection.createStatement();
                    ResultSet res11 = conn.executeQuery(query11);
                    System.out.println("Emp_SSN\t | Total_Expense\t");
                    System.out.println("---------------------------------");
                    while (res11.next()) {
```


shehnazshaik@ou.edu

```
                    System.out.println(res11.getString(1) + " | " +
res11.getString(2));
                    }
                    break;
                //12) Retrieve the list of volunteers that are members of teams that
support a particular client
                case 12:
                    System.out.println("Retrieving All the Client SSN to ultimately
retrieve volunteers that are members of teams\n");
                    final String rquery = "Select SSN FROM Cares;";
                    conn = connection.createStatement();
                    ResultSet rres = conn.executeQuery(rquery);
                    while(rres.next()) {
                        System.out.println(rres.getString(1));
                        }
                    System.out.println("Enter any client SSN from above
results:\n");
                    String cl_ssn = in.next();
                    final String query12 = "SELECT SSN, Name FROM Person WHERE
SSN in (SELECT SSN FROM Serve WHERE T_name in (SELECT T_name FROM Cares WHERE SSN='"+ cl_ssn
+"'))";
                    conn = connection.createStatement();
                    ResultSet res12 = conn.executeQuery(query12);
                    System.out.println("Retrieving Volunteers:\n");
                    System.out.println("SSN\t |\tName");
                    System.out.println("--------------");
                    while (res12.next()) {
                    System.out.println(res12.getString(1) +"|" + res12.getString(2));
                    }
                    System.out.println("Query 12 suuccessfully executed");
                    break;
```


shehnazshaik@ou.edu

```
                    //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
                    case 13:
                        final String query13 = "SELECT Name, Address, email, home_Num, work_Num, cell_Num FROM Person WHERE SSN in(SELECT SSN FROM Cares WHERE T_Name in(SELECT T_Name FROM Sponsor WHERE O_Name BETWEEN 'B%' AND 'K%'));";
                        conn = connection.createStatement();
                        ResultSet res13 = conn.executeQuery(query13);
                        System.out.println("Name \t| Address \t| email \t| home_Num \t| work_Num \t| cell_Num\t");
                        System.out.println("----------------------------------------------------------------------------------");
                        while (res13.next()) {
                            System.out.println(res13.getString(1)+" |"+res13.getString(2)+" \t|"+res13.getString(3)+" |"+res13.getString(4)+"   |"+res13.getString(5)+" |"+res13.getString(6));
                            }
                        break;
                    //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
                    case 14:
                        final String query14 = "SELECT P.Name, sum(DO.D_Amount), D.isAnonymous as DO_Amt\n" +
                                "FROM Donor D, Employee E, Person P,\n" +
                                "(SELECT SSN, D_Amount from CreditCard\n" +
                                "UNION ALL \n" +
                                "SELECT SSN, D_Amount from Cheque) as DO\n" +
                                "WHERE P.SSN   = D.SSN\n" +
                                "AND E.SSN = D.SSN\n" +
```


shehnazshaik@ou.edu

```
                                                        "AND DO.SSN = D.SSN\n" +
                                                        "GROUP BY D.SSN, P.Name,
D.isAnonymous ORDER BY DO_Amt;";

                                        conn = connection.createStatement();
                                        ResultSet res14 = conn.executeQuery(query14);
                                        System.out.println("Retriving Donors: If the Donor is
Anonymous you will see 'Anonymous' as his or her name\n");
                                        System.out.println("Name \t| Tot_Amount \t| isAnonymous
\t|");
                                        System.out.println("----------------------------------------");
                                        String Anonymous;
                                        String A_Status;
                                        String Donor_Name;
                                        while (res14.next()) {
                                                Anonymous = res14.getString(3);
                                                A_Status = res14.getString(3);
                                                if(Anonymous == "1") {
                                                        Donor_Name = "Anonymous";
                                                        A_Status = "True";
                                                }
                                                else {
                                                        Donor_Name = res14.getString(1);
                                                        A_Status = "False";
                                                }
                                                System.out.println(Donor_Name+"\t|"+
res14.getString(2) +"\t|"+ A_Status);
                                                }
                                        System.out.println("Query 14 suuccessfully executed");
                                        break;
```


shehnazshaik@ou.edu

```
                    //15) Retrieve the names of all teams that were founded after a
particular date
                    case 15:
                        System.out.println("Enter the Date (DD-Mon-YYYY):\n");
                        String f_date = in.next();
                        final String query15 = "SELECT T_Name FROM Teams WHERE
T_FormDate > '"+ f_date +"';";
                        conn = connection.createStatement();
                        ResultSet res15 = conn.executeQuery(query15);
                        System.out.println("Retrieving Team Names formed after above
date");
                        System.out.println("Team_Name ");
                        System.out.println("-----------");
                        while (res15.next()) {
                            System.out.println(res15.getString(1));
                            }
                        System.out.println("Query 15 suuccessfully executed");
                        break;
                    //16) Increase the salary by 10% of all employees to whom more than
one team must report.
                    case 16:
                        final String query16 = "UPDATE Employee SET Salary =
(Salary*1.1) WHERE SSN IN (SELECT SSN FROM Report GROUP BY SSN Having count(SSN) > 1);";
                        conn = connection.createStatement();
                        conn.execute(query16);
                        System.out.println("Updated Employee Table");
                        final String fquery16 = "SELECT SSN, Salary FROM Employee;";
                        conn = connection.createStatement();
                        ResultSet fres16 = conn.executeQuery(fquery16);
                        System.out.println("Retrieving the Employee SSN and salary to
check if update has been made");
```


shehnazshaik@ou.edu

```
                    System.out.println("Emp_SSN \t| Salary \t|");
                    System.out.println("--------------------------");
                    while (fres16.next()) {
                        System.out.println(fres16.getString(1)+"    |"+
fres16.getString(2));
                        }
                    System.out.println("Query 16 suuccessfully executed");
                    break;
                //17) Delete all clients who do not have health insurance and whose
value of importance for transportation is less than 5
                case 17:
                    final String query17 = "DELETE FROM Client WHERE SSN NOT
IN(SELECT SSN FROM Insurance WHERE Policy_Type = 'Health') AND SSN IN (SELECT SSN FROM Needs
WHERE Need_Type = 'Transportation' AND Quantity < 5)";
                    conn = connection.createStatement();
                    conn.executeQuery(query17);
                    System.out.println("Record(s) Deleted");
                    System.out.println("Checking if the records are deleted");
                    final String fquery17 = " SELECT SSN FROM Client;";
                    conn = connection.createStatement();
                    ResultSet fres17 = conn.executeQuery(fquery17);
                    System.out.println("Client_SSN \t");
                    System.out.println("------------------");
                    while (fres17.next()) {
                        System.out.println(fres17.getString(1));
                        }
                    System.out.println("Query 17 suuccessfully executed");
                    break;
                //18) Import: enter new teams from a data file until the file is empty
(the user must be asked to enter the input file name).
```


shehnazshaik@ou.edu

```
                    case 18:
                        System.out.println("Enter the file name: ");
                            String fileName = in.next();
                            try {
                                //File reader in tha following path
                                FileInputStream fs = new
FileInputStream("C:\\MS\\4513\\IndividualProject\\"+fileName);
                                DataInputStream ds = new DataInputStream(fs);
                                BufferedReader br = new BufferedReader(new
InputStreamReader(ds));
                                String s;
                                //reading every line
                                while ((s = br.readLine()) != null)
                                {
                                    String i[] = s.split(","); //splitting each
record with a comma to feed into insert query

                                    conn = connection.createStatement();
                                    conn.executeUpdate("INSERT INTO
Teams VALUES('"+i[0]+ "','" +i[1]+ "', '"+i[2]+"')");
                                }
                                br.close();
                                }
                                catch (FileNotFoundException FNFE) {
                                    FNFE.printStackTrace();
                                }
                                catch (IOException IOE){
                                    IOE.printStackTrace();
                                }
                                System.out.println("File has been successfuly
Imported!\n");
```


shehnazshaik@ou.edu

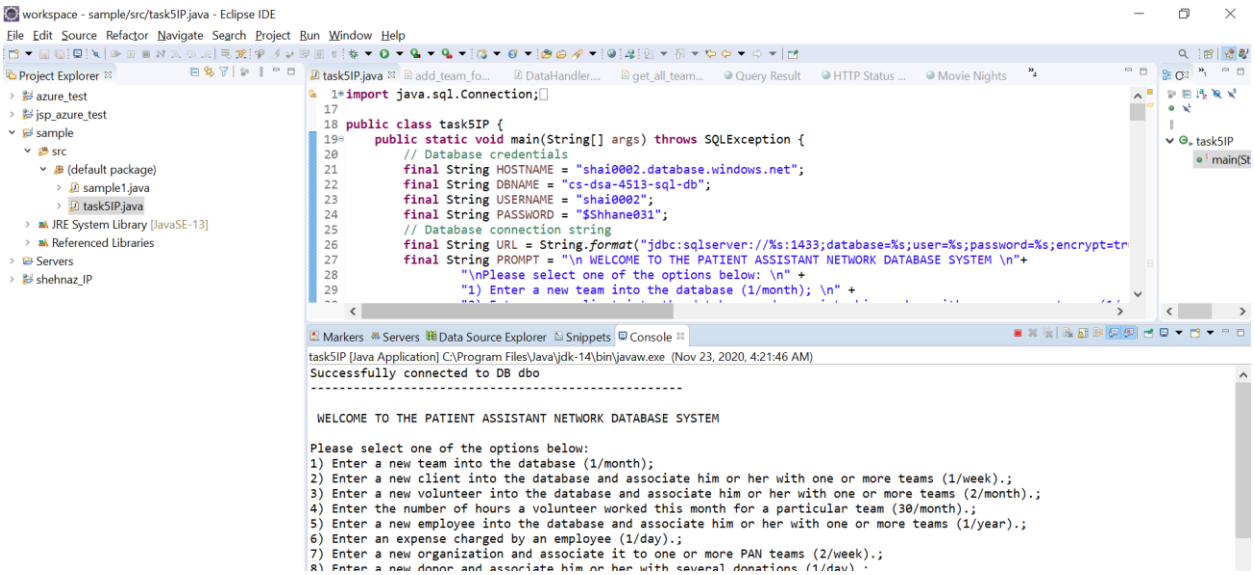
```
                    break;
                //19) Export: Retrieve names and mailing addresses of all people on the mailing list and output them to a data file instead of screen (the user must be asked to enter the output file name).
                case 19:
                    System.out.println("Enter the name of the file to be exported\n");
                        String outfile = in.next();
                        final String query19 = "SELECT Name, Address FROM Person WHERE is_inMailList = 1;";
                        conn = connection.createStatement();
                        ResultSet res19 = conn.executeQuery(query19);
                        try {
                            //creating a writer in the specified path.
                            BufferedWriter op = new BufferedWriter(new FileWriter("C:/MS/4513/IndividualProject/"+outfile));
                            //writing every record with comma separation
                            while(res19.next()){
                                op.write("\n");
                                op.write(res19.getString(1)+",\t");
                                op.write(res19.getString(2)+"\n");
                                }
                            op.close();
                        }
                        catch(FileNotFoundException FOE) {
                            FOE.printStackTrace();
                        }
                        catch (IOException IOE){
                            IOE.printStackTrace();
                        }
```

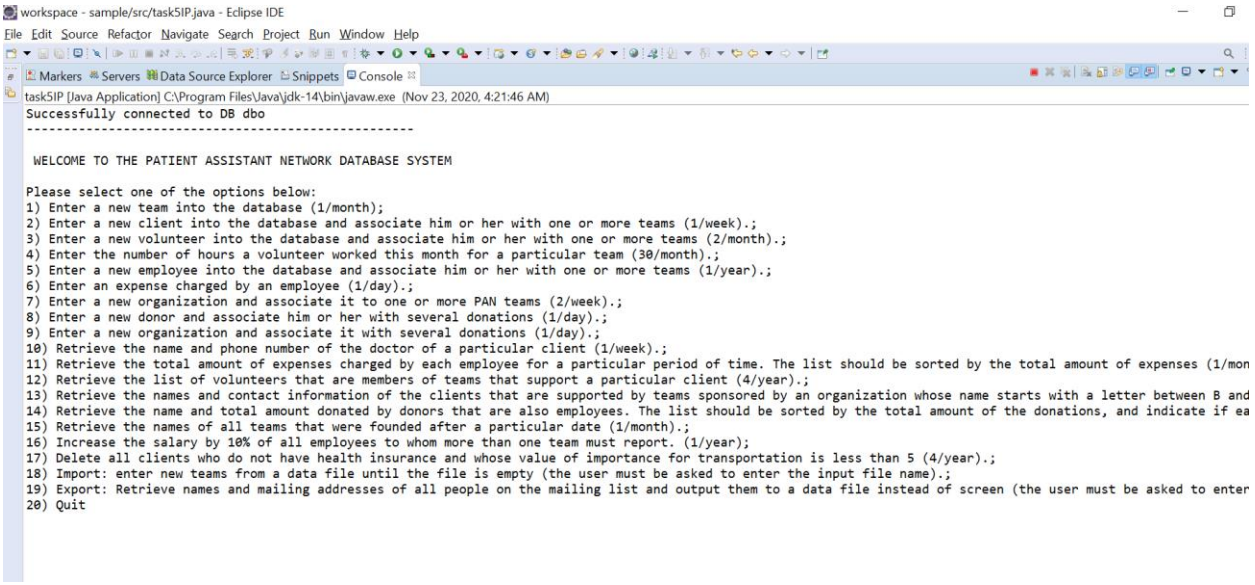

shehnazshaik@ou.edu

```
                                        System.out.println("export successful\n");
                                    break;
                                case 20:
                                        in.close();
                                    System.out.println("Quit. \n Exiting... \n Goodbye!");
                                    System.exit(0);
                            break;
                        default: // Unrecognized option, re-prompt the user for the correct one
                            System.out.println(String.format(
                                                "Unrecognized option: %s\n" +
                               "Please try again!",
                               option));
                            break;
                                    }


                                }
                    }
                    catch(SQLException e) {
                            System.out.println("We have an error: " + e.getMessage());
                    }
            }
}
```

Successful execution of Java Project:

shehnazshaik@ou.edu





Task 6. Execution of the Java Program

6. 1 Testing query 1

PAN database Prompt screenshot:

shehnazshaik@ou.edu

```
task5IP [Java Application]
Successfully connected to DB dbo
-----------------------------------------------------

 WELCOME TO THE PATIENT ASSISTANT NETWORK DATABASE SYSTEM

Please select one of the options below:
1) Enter a new team into the database (1/month);
2) Enter a new client into the database and associate him or her with one or more teams (1/week).;
3) Enter a new volunteer into the database and associate him or her with one or more teams (2/month).;
4) Enter the number of hours a volunteer worked this month for a particular team (30/month).;
5) Enter a new employee into the database and associate him or her with one or more teams (1/year).;
6) Enter an expense charged by an employee (1/day).;
7) Enter a new organization and associate it to one or more PAN teams (2/week).;
8) Enter a new donor and associate him or her with several donations (1/day).;
9) Enter a new organization and associate it with several donations (1/day).;
10) Retrieve the name and phone number of the doctor of a particular client (1/week).;
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 (1/mont
12) Retrieve the list of volunteers that are members of teams that support a particular client (4/year).;
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
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 eac
15) Retrieve the names of all teams that were founded after a particular date (1/month).;
16) Increase the salary by 10% of all employees to whom more than one team must report. (1/year);
17) Delete all clients who do not have health insurance and whose value of importance for transportation is less than 5 (4/year).;
18) Import: enter new teams from a data file until the file is empty (the user must 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 must be asked to enter
20) Quit
```

Option1 Query Execution:

Run 1:

```
20) Quit
1
Enter the Team Name:

RCB
Enter the Team Type:

IPL

Enter the Team Form date (DD-Mon-YYYY):

01-Apr-2013

1 record successfully inserted
```

Run 2:

```
1
Enter the Team Name:

Bears
Enter the Team Type:

NFL
Enter the Team Form date (DD-Mon-YYYY):

15-Feb-2012
1 record successfully inserted
```

Run 3:

```
1
Enter the Team Name:

CSK
Enter the Team Type:

IPL
Enter the Team Form date (DD-Mon-YYYY):

30-May-2008
1 record successfully inserted
```


shehnazshaik@ou.edu

Run 4:

```
1
Enter the Team Name:

OU
Enter the Team Type:

NCAA
Enter the Team Form date (DD-Mon-YYYY):

14-Dec-2010
1 record successfully inserted
```

Run 5:

```
1
Enter the Team Name:

OKC Thunder
Enter the Team Type:

NBA
Enter the Team Form date (DD-Mon-YYYY):

14-Dec-2010
1 record successfully inserted
```

Records reflecting in DB:

```
1   SELECT * FROM Teams;
```

Results Messages

	T_Name	T_Type	T_FormDate
1	Bears	NFL	2012-02-15
2	CSK	IPL	2008-05-30
3	OKC Thunder	NBA	2010-12-14
4	OU	NCAA	2010-12-14
5	RCB	IPL	2013-04-01

6. 2 Testing query 2

Option 2 Query Execution:

Run1

shehnazshaik@ou.edu

```
2
Enter values for Option 2

Enter SSN of new Client

4513001
Enter the Doctor Name:

Karlo
Enter the Attorney Name:

Maya
Enter the Doctor Number:

4051112221
Enter the Attorney Number:

4051112221
Enter the Organization assigned date (DD-Mon-YYYY):

01-Feb-2017
New SSN Added to Client

Enter i : No of Teams, newly added client is associated with:

1
Enter the Team Name:

OU
Enter Team Care Status:

Active
Query 2 suuccessfully executed
```

Run 2:

```
2
Enter values for Option 2

Enter SSN of new Client

4513005
Enter the Doctor Name:

Jarvis
Enter the Attorney Name:

Bruce
Enter the Doctor Number:

8095702567
Enter the Attorney Number:

8095789567
Enter the Organization assigned date (DD-Mon-YYYY):

21-Sep-2018
New SSN Added to Client

Enter i : No of Teams, newly added client is associated with:

1
Enter the Team Name:

Bears
Enter Team Care Status:

InActive
Query 2 suuccessfully executed
```

Run 3:

shehnazshaik@ou.edu

```
2
Enter values for Option 2

Enter SSN of new Client

4513002
Enter the Doctor Name:

Jarvis
Enter the Attorney Name:

Bruce
Enter the Doctor Number:

8095789567
Enter the Attorney Number:

8095700567
Enter the Organization assigned date (DD-Mon-YYYY):

31-Aug-2018
New SSN Added to Client

Enter i : No of Teams, newly added client is associated with:

1
Enter the Team Name:

CSK
Enter Team Care Status:

Active
Query 2 suuccessfully executed
```

Run 4:

shehnazshaik@ou.edu

```
2
Enter values for Option 2

Enter SSN of new Client

4513007
Enter the Doctor Name:

Kang
Enter the Attorney Name:

Henry
Enter the Doctor Number:

8095769567
Enter the Attorney Number:

8095340567
Enter the Organization assigned date (DD-Mon-YYYY):

19-Jan-2017
New SSN Added to Client

Enter i : No of Teams, newly added client is associated with:

1
Enter the Team Name:

OU
Enter Team Care Status:

Active
Query 2 suuccessfully executed
```

Run 5:

shehnazshaik@ou.edu

```
2
Enter values for Option 2

Enter SSN of new Client

4513012
Enter the Doctor Name:

Mike
Enter the Attorney Name:

Mindy
Enter the Doctor Number:

8095749500
Enter the Attorney Number:

8095723567
Enter the Organization assigned date (DD-Mon-YYYY):

04-Oct-2019
New SSN Added to Client

Enter i : No of Teams, newly added client is associated with:

1
Enter the Team Name:

SRH
Enter Team Care Status:

InActive
Query 2 suuccessfully executed
```

Records Inserted in DB:

```
1  SELECT * FROM Client;
2  SELECT * FROM Cares;
```

Results | Messages

	SSN	Doc_Name	Att_Name	Doc_Number	Att_Number	Org_AssignDate
1	4513001	Karlo	Maya	4051112221	4051112221	2017-02-01
2	4513002	Jarvis	Bruce	8095789567	8095700567	2018-08-31
3	4513005	Jarvis	Bruce	8095702567	8095789567	2018-09-21
4	4513007	Kang	Henry	8095769567	8095340567	2017-01-19
5	4513012	Mike	Mindy	8095749500	8095723567	2019-04-10

	SSN	T_Name	Status
1	4513001	OU	Active
2	4513002	CSK	Active
3	4513005	Bears	InActive
4	4513007	OU	InActive
5	4513012	SRH	InActive

shehnazshaik@ou.edu

6. 3 Testing query 3

Option3 Query Execution:

Run 1: Assigning 2 Teams

```
3
Enter values for Option 3

Enter SSN of new Volunteer
4513007
Enter the Volunteer Joining date (DD-Mon-YYYY):
12-Mar-2017
Enter the Volunteer Training date (DD-Mon-YYYY):
17-Sep-2017
Enter the Volunteer Training Location:
Dallas
Enter i : No of Teams, newly added Volunteer serves:

2
Enter the Team Name:

OU
Enter the No. of Hours:

20
Enter the Name of the Month:

September
Enter volunteer Status:

InActive
Enter the Team Name:

CSK


CSK
Enter the No. of Hours:

50
Enter the Name of the Month:

October
Enter volunteer Status:

Active
Query 3 suuccessfully executed
```

Run 2:

shehnazshaik@ou.edu

```
3
Enter values for Option 3

Enter SSN of new Volunteer
4513001
Enter the Volunteer Joining date (DD-Mon-YYYY):
28-Feb-2018
Enter the Volunteer Training date (DD-Mon-YYYY):
29_Mar-2020
Enter the Volunteer Training Location:
OKC
Enter i : No of Teams, newly added Volunteer serves:

1
Enter the Team Name:

OKC Thunder
Enter the No. of Hours:

60
Enter the Name of the Month:

August
Enter volunteer Status:

Active
Query 3 suuccessfully executed
```

Run 3:

```
3
Enter values for Option 3

Enter SSN of new Volunteer
4513002
Enter the Volunteer Joining date (DD-Mon-YYYY):
17-Mar-2016
Enter the Volunteer Training date (DD-Mon-YYYY):
17-Jul-2018
Enter the Volunteer Training Location:
Tulsa
Enter i : No of Teams, newly added Volunteer serves:

1
Enter the Team Name:

OU
Enter the No. of Hours:

40
Enter the Name of the Month:

August
Enter volunteer Status:

Active
Query 3 suuccessfully executed
```

Run 4:

shehnazshaik@ou.edu

```
3
Enter values for Option 3

Enter SSN of new Volunteer
4513005
Enter the Volunteer Joining date (DD-Mon-YYYY):
30-Sep-2017
Enter the Volunteer Training date (DD-Mon-YYYY):
13-Jun-2019
Enter the Volunteer Training Location:
Michigan
Enter i : No of Teams, newly added Volunteer serves:

1
Enter the Team Name:

CSK
Enter the No. of Hours:

15
Enter the Name of the Month:

May
Enter volunteer Status:

InActive
Query 3 suuccessfully executed
```

Run 5:

```
3
Enter values for Option 3

Enter SSN of new Volunteer
4513011
Enter the Volunteer Joining date (DD-Mon-YYYY):
31-May-2015
Enter the Volunteer Training date (DD-Mon-YYYY):
15-Aug-2017
Enter the Volunteer Training Location:
Tampa
Enter i : No of Teams, newly added Volunteer serves:

1
Enter the Team Name:

Bears
Enter the No. of Hours:

50
Enter the Name of the Month:

September
Enter volunteer Status:

InActive
Query 3 suuccessfully executed
```

Records Reflecting in DB:

shehnazshaik@ou.edu

```
SELECT * FROM Volunteer;
SELECT * FROM Serve;
```

Results Messages

	SSN	V_DOJ	Training_Date	Traing_Loc
1	4513001	2018-02-28	2020-03-29	OKC
2	4513002	2016-03-17	2018-07-17	Tulsa
3	4513005	2017-09-30	2019-06-13	Michigan
4	4513007	2017-03-12	2017-09-17	Dallas
5	4513011	2015-05-31	2017-08-15	Tampa

	SSN	T_Name	Hours	Month	V_Status
1	4513001	OKC Thunder	60	August	Active
2	4513002	OU	40	August	Active
3	4513005	CSK	15	May	InActive
4	4513007	CSK	50	October	Active
5	4513007	OU	20	September	InActive
6	4513011	Bears	50	September	InActive

6. 4 Testing query 4

Option 4 Query Execution:

Run 1:

```
4
Enter SSN of a Volunteer

4513005
Enter the Team Name:

RCB
Enter the No. of Hours:

88
Enter the Name of the Month:

November
Enter volunteer Status:

Active
Query  suuccessfully executed
```

Run 2:

shehnazshaik@ou.edu

```
4
Enter SSN of a Volunteer

4513001
Enter the Team Name:

OU
Enter the No. of Hours:

15
Enter the Name of the Month:

November
Enter volunteer Status:

InActive
Query  suuccessfully executed
```

Run 3:

```
4513007
Enter the Team Name:

SRH
Enter the No. of Hours:

40
Enter the Name of the Month:

November
Enter volunteer Status:

InActive
Query  suuccessfully executed
```

Run 4:

```
4513011
Enter the Team Name:

CSK
Enter the No. of Hours:

60
Enter the Name of the Month:

November
Enter volunteer Status:

Active
Query  suuccessfully executed
```

Run 5:

shehnazshaik@ou.edu

```
4
Enter SSN of a Volunteer

4513002
Enter the Team Name:

Bears
Enter the No. of Hours:

80
Enter the Name of the Month:

November
Enter volunteer Status:

Active
Query  suuccessfully executed
```

Records are reflecting in DB:

```
1    SELECT * FROM Volunteer;
2    SELECT * FROM Serve;
```

Results Messages

	SSN	T_Name	Hours	Month	V_Status
1	4513001	OKC Thunder	60	August	Active
2	4513001	OU	15	November	InActive
3	4513002	Bears	80	November	Active
4	4513002	OU	40	August	Active
5	4513005	CSK	15	May	InActive
6	4513005	RCB	88	November	Active
7	4513007	CSK	50	October	Active
8	4513007	OU	20	September	InActive
9	4513007	SRH	40	November	InActive
1…	4513011	Bears	50	September	InActive
1…	4513011	CSK	60	November	Active

6.5 Testing query 5

Option 5 Query Execution:

Run 1:

shehnazshaik@ou.edu

```
5
Enter values for Option 5

Enter SSN of new Employee

4513009
Enter the Salary:

78000
Enter the Marital Status:

Single
Enter the Employee Hire date (DD-Mon-YYYY):

25-Jul-2017
Enter i : No of Teams, newly added Employee is Associated to:

1
Enter the Team Name:

OKC Thunder
Enter the report date (DD-Mon-YYYY):

30-Jun-2019
Enter report description:

Checks marked
Query 5 suuccessfully executed
```

Run 2:

```
5
Enter values for Option 5

Enter SSN of new Employee

4513006
Enter the Salary:

85000
Enter the Marital Status:

Widowed
Enter the Employee Hire date (DD-Mon-YYYY):

28-May-2015
Enter i : No of Teams, newly added Employee is Associated to:

1
Enter the Team Name:

Bears
Enter the report date (DD-Mon-YYYY):

30-Jun-2019
Enter report description:

Fitness Check
Query 5 suuccessfully executed
```

Run 3:

shehnazshaik@ou.edu

```
5
Enter values for Option 5

Enter SSN of new Employee

4513007
Enter the Salary:

73000
Enter the Marital Status:

Single
Enter the Employee Hire date (DD-Mon-YYYY):

10-Oct-2016
Enter i : No of Teams, newly added Employee is Associated to:

1
Enter the Team Name:

OU
Enter the report date (DD-Mon-YYYY):

31-Jul-2019
Enter report description:

Strategy upadate
Query 5 suuccessfully executed
```

Run 4:

```
5
Enter values for Option 5

Enter SSN of new Employee

4513003
Enter the Salary:

67000
Enter the Marital Status:

Married
Enter the Employee Hire date (DD-Mon-YYYY):

13-Mar-2018
Enter i : No of Teams, newly added Employee is Associated to:

2
Enter the Team Name:

CSK
Enter the report date (DD-Mon-YYYY):

31-May-2019
Enter report description:

Passed
Enter the Team Name:
```


shehnazshaik@ou.edu

```
SRH
Enter the report date (DD-Mon-YYYY):

31-Aug-2019
Enter report description:

Practice-done
Query 5 suuccessfully executed
```

Run 5:

```
5
Enter values for Option 5

Enter SSN of new Employee

4513012
Enter the Salary:

70000
Enter the Marital Status:

Divorced
Enter the Employee Hire date (DD-Mon-YYYY):

29-Dec-2016
Enter i : No of Teams, newly added Employee is Associated to:

1
Enter the Team Name:

Bears
Enter the report date (DD-Mon-YYYY):

30-Jun-2017
Enter report description:

Met All Expectations
Query 5 suuccessfully executed
```

Records reflecting in the db:

shehnazshaik@ou.edu

```
1   SELECT * FROM Employee;
2   SELECT * FROM Report;
```

Results Messages

	SSN	Salary	Marital_Status	Hire_Date
1	4513003	73700	Married	2018-03-13
2	4513006	85000	Widowed	2015-05-28
3	4513007	73000	Single	2016-10-10
4	4513009	78000	Single	2017-07-25
5	4513011	80000	Married	2016-11-28
6	4513012	70000	Divorced	2016-12-29

	SSN	T_Name	R_Date	R_Desc
1	4513003	CSK	2019-05-31	Passed
2	4513003	SRH	2019-08-31	Practice-done
3	4513006	Bears	2019-06-30	Fitness check
4	4513007	OU	2019-07-31	Strategy upadate
5	4513009	OKC Thunder	2019-06-30	Checks marked
6	4513012	Bears	2017-06-30	Met All Expectations

6.6 Testing query 6

Option 6 Query Execution:

Run 1:

```
6
Enter SSN of an Employee

4513006
Enter the Expense description:

Meals
Enter the expense Amount:

300
Enter the expense date (DD-Mon-YYYY)::

15-Nov-2019
Query 6 suuccessfully executed
```

Run 2:

```
6
Enter SSN of an Employee

4513003
Enter the Expense description:

Water Bottles
Enter the expense Amount:

150
Enter the expense date (DD-Mon-YYYY)::

15-May-2019
Query 6 suuccessfully executed
```


shehnazshaik@ou.edu

Run 3:

```
6
Enter SSN of an Employee

4513007
Enter the Expense description:

Food
Enter the expense Amount:

1000
Enter the expense date (DD-Mon-YYYY)::

15-Jun-2019
Query 6 suuccessfully executed
```

Run 4:

```
6
Enter SSN of an Employee

4513009
Enter the Expense description:

Travel
Enter the expense Amount:

2500
Enter the expense date (DD-Mon-YYYY)::

15-Jul-2019
Query 6 suuccessfully executed
```

Run 5:

```
6
Enter SSN of an Employee

4513011
Enter the Expense description:

Energy Drinks
Enter the expense Amount:

500
Enter the expense date (DD-Mon-YYYY)::

15-Aug-2019
Query 6 suuccessfully executed
```

Records reflecting in DB:

```
1   SELECT * FROM Expenses;
```

Results | Messages

	SSN	E_Desc	E_Amount	E_Date
1	4513003	Water Bottles	150	2019-05-15
2	4513006	Meals	300	2019-11-15
3	4513007	Food	1000	2019-06-15
4	4513009	Travel	2500	2019-07-15
5	4513011	Energy Drinks	500	2019-08-15

shehnazshaik@ou.edu

6. 7 Testing query 7

Option 7 Query Execution:

Run 1:

```
7
Enter values for Option 7

Enter Name of new Organization

Lindt
Enter the mail address:

E Brooks
Enter the Phone Number:

4051112224
Enter the Contact Person

Rafal
Enter i : No of Teams, newly added Organization sponsors:

1
Enter the Team Name:

RCB
Query 7 suuccessfully executed
```

Run 2:

```
7
Enter values for Option 7

Enter Name of new Organization

BPF
Enter the mail address:

SanDiego
Enter the Phone Number:

8095741110
Enter the Contact Person

Chris
Enter i : No of Teams, newly added Organization sponsors:

2
Enter the Team Name:

OU
Enter the Team Name:

SRH
Query 7 suuccessfully executed
7
```

Run 3:

shehnazshaik@ou.edu

```
7
Enter values for Option 7

Enter Name of new Organization

Carmax
Enter the mail address:

Richmond
Enter the Phone Number:

8095741111
Enter the Contact Person

Steve
Enter i : No of Teams, newly added Organization sponsors:

1
Enter the Team Name:

SRH
Query 7 suuccessfully executed
```

Run 4:

```
7
Enter values for Option 7

Enter Name of new Organization

Philips
Enter the mail address:

Houston
Enter the Phone Number:

8095740001
Enter the Contact Person

Jim
Enter i : No of Teams, newly added Organization sponsors:

1
Enter the Team Name:

Bears
Query 7 suuccessfully executed
```

Run 5:

shehnazshaik@ou.edu

```
7
Enter values for Option 7

Enter Name of new Organization

TMobile
Enter the mail address:

Vegas
Enter the Phone Number:

8095749001
Enter the Contact Person

Jeff
Enter i : No of Teams, newly added Organization sponsors:

2
Enter the Team Name:

OU
Enter the Team Name:

SRH
Query 7 suuccessfully executed
```

Records reflecting in DB:

```
1  SELECT * FROM Ext_Organization;
2  SELECT * FROM Sponsor;
```

Results Messages

	O_Name	O_mail_Address	O_Phone_Num	O_ContactPerson
1	BPF	SanDiego	8095741110	Chris
2	Carmax	Richmond	8095741111	Steve
3	Lindt	E Brooks	4051112224	Rafal
4	Philips	Houston	8095740001	Jim
5	TMobile	Vegas	8095749001	Jeff

	O_Name	T_Name
1	BPF	OU
2	BPF	SRH
3	Carmax	SRH
4	Lindt	RCB
5	Philips	Bears
6	TMobile	OU
7	TMobile	SRH

6. 8 Testing query 8

Option 8 Query Execution:

shehnazshaik@ou.edu

Run 1:

```
8
Enter values for Option 8

Enter the SSN of new Donor

4513006
Enter  "1" if donor is Anonymous, "0"  if not:

1
Enter i : No of Donations, newly added Donor makes:

2
Enter "1" for Cheque and "2" for Creditcard donation

1
Enter the date of donation (DD-Mon-YYYY)::

31-Jan-2019
Enter the Donation Amount:

300
Enter the Donation Type:

Savings
Enter the Donation Campaign Name:

Enter the Donation Campaign Name:

OCD1
Enter the Cheque Number:

123401
Enter "1" for Cheque and "2" for Creditcard donation

2
Enter the date of donation (DD-Mon-YYYY)::

01-Jan-2019
Enter the Donation Amount:

300
Enter the Donation Type:

Savings
Enter the Donation Campaign Name:

OCD1
Enter the Credit card Number:

9723899893
Enter the Card Type:

Visa
Enter the Card Expiry date (DD-Mon-YYYY)::

28-Feb-2021
Query 8 suuccessfully executed
```

Run 2:

shehnazshaik@ou.edu

```
8
Enter values for Option 8

Enter the SSN of new Donor

4513012
Enter  "1" if donor is Anonymous, "0"  if not:

0
Enter i : No of Donations, newly added Donor makes:

2
Enter "1" for Cheque and "2" for Creditcard donation

1
Enter the date of donation (DD-Mon-YYYY)::

28-Feb-2019
Enter the Donation Amount:

150
Enter the Donation Type:

Fund
Enter the Donation Campaign Name:

OCD2
Enter the Cheque Number:
```

```
Enter the Cheque Number:

123402
Enter "1" for Cheque and "2" for Creditcard donation

2
Enter the date of donation (DD-Mon-YYYY)::

02-Feb-2019
Enter the Donation Amount:

170
Enter the Donation Type:

Fund
Enter the Donation Campaign Name:

OCD2
Enter the Credit card Number:

9723899892
Enter the Card Type:

Discover
Enter the Card Expiry date (DD-Mon-YYYY)::

28-Feb-2022
Query 8 suuccessfully executed
```

Run 3:

shehnazshaik@ou.edu

```
8
Enter values for Option 8

Enter the SSN of new Donor

4513010
Enter  "1" if donor is Anonymous, "0"  if not:

0
Enter i : No of Donations, newly added Donor makes:

1
Enter "1" for Cheque and "2" for Creditcard donation

1
Enter the date of donation (DD-Mon-YYYY)::

31-Mar-2019
Enter the Donation Amount:

200
Enter the Donation Type:

Savings
Enter the Donation Campaign Name:

OCD3
Enter the Cheque Number:

123403
Query 8 suuccessfully executed
```

Run 4:

```
8
Enter values for Option 8

Enter the SSN of new Donor

4513008
Enter  "1" if donor is Anonymous, "0"  if not:

1
Enter i : No of Donations, newly added Donor makes:

1
Enter "1" for Cheque and "2" for Creditcard donation

2
Enter the date of donation (DD-Mon-YYYY)::

'01-Mar-2019'
Enter the Donation Amount:

500
Enter the Donation Type:

Savings
Enter the Donation Campaign Name:
```


shehnazshaik@ou.edu

```
Enter the Donation Campaign Name:

OCD3
Enter the Credit card Number:

9723899894
Enter the Card Type:

Visa
Enter the Card Expiry date (DD-Mon-YYYY)::

28-Feb-2023
Query 8 suuccessfully executed
```

Run 5:

```
8
Enter values for Option 8

Enter the SSN of new Donor

4513001
Enter  "1" if donor is Anonymous, "0"  if not:

0
Enter i : No of Donations, newly added Donor makes:

1
Enter "1" for Cheque and "2" for Creditcard donation

1
Enter the date of donation (DD-Mon-YYYY)::

30-Jun-2019
Enter the Donation Amount:

200
Enter the Donation Type:

Savings
Enter the Donation Campaign Name:

OCD4
Enter the Cheque Number:

123404
Query 8 suuccessfully executed
```


shehnazshaik@ou.edu

```
SELECT * FROM Donor;
SELECT * FROM Cheque;
SELECT * FROM CreditCard;
```

Results Messages

	SSN	isAnonymous
1	4513001	0
2	4513006	1
3	4513008	1
4	4513010	0
5	4513012	0

	SSN	D_Date	D_Amount	D_Type	D_Camp_Name	ChequeNum
1	4513001	2019-06-30	200	Savings	OCD4	123404
2	4513006	2019-01-31	300	Savings	OCD1	123401
3	4513010	2019-03-31	200	Savings	OCD3	123403
4	4513012	2019-02-28	150	Fund	OCD2	123402

	SSN	D_Date	D_Amount	D_Type	D_Camp_Name	Card_Num	Card_Type	Exp_Type
1	4513006	2019-01-01	300	Savings	OCD1	9723899893	Visa	2021-02-28
2	4513006	2019-02-02	170	Fund	OCD2	9723899892	Discover	2022-02-28
3	4513008	2019-03-01	500	Savings	OCD3	9723899894	Visa	2023-02-28
4	4513012	2019-02-02	170	Fund	OCD2	9723899892	Discover	2022-02-28

Task 6. 9 Testing query 9

Option 9 Query Execution:

Run 1:

shehnazshaik@ou.edu

```
9
Enter values for Option 9

Enter Name of new Organization

CVS
Enter the mail address:

W Main
Enter the Phone Number:

4051112224
Enter the Contact Person

Sam
Organization inserted suuccessfully
Enter i : No of Donations, newly added Organization makes:

2
Enter "1" for Cheque and "2" for Creditcard donation

1
Enter the date of donation (DD-Mon-YYYY)::

01-Jan-2020
Enter the Donation Amount:

550
Enter the Donation Type:

Funds

Funds
Enter the Donation Campaign Name:

Camp1
Enter  "1" if Org is Anonymous, "0"  if not:

0
Enter the Cheque Number:

101120
Enter "1" for Cheque and "2" for Creditcard donation

2
Enter the date of donation (DD-Mon-YYYY)::

02-Jan-2020
Enter the Donation Amount:

550
Enter the Donation Type:

Compfund
Enter the Donation Campaign Name:

Camp2
Enter  "1" if Org is Anonymous, "0"  if not:

0
```


shehnazshaik@ou.edu

```
Enter  "1" if Org is Anonymous, "0"  if not:

0
Enter the Credit card Number:

1234121310
Enter the Card Type:

Visa
Enter the Card Expiry date (DD-Mon-YYYY)::

23-Jun-2022
Query 9 suuccessfully executed
```

Run 2:

```
9
Enter values for Option 9

Enter Name of new Organization

Pepsico
Enter the mail address:

W Elhm
Enter the Phone Number:

4053334445
Enter the Contact Person

Chris
Organization inserted suuccessfully
Enter i : No of Donations, newly added Organization makes:

1
Enter "1" for Cheque and "2" for Creditcard donation

2
Enter the date of donation (DD-Mon-YYYY)::

03-Jan-2020
Enter the Donation Amount:

1000
Enter the Donation Type:

CompF
Enter the Donation Campaign Name:
```

```
Enter the Donation Campaign Name:

Camp3
Enter  "1" if Org is Anonymous, "0"  if not:

0
Enter the Credit card Number:

123455431
Enter the Card Type:

Discover
Enter the Card Expiry date (DD-Mon-YYYY)::

25-Jun-2023
Query 9 suuccessfully executed
```

Run 3:

shehnazshaik@ou.edu

```
9
Enter values for Option 9

Enter Name of new Organization

ABC
Enter the mail address:

W Memorial
Enter the Phone Number:

4056667771
Enter the Contact Person

Pratt
Organization inserted suuccessfully
Enter i : No of Donations, newly added Organization makes:

1
Enter "1" for Cheque and "2" for Creditcard donation

1

Enter the date of donation (DD-Mon-YYYY)::

04-Jan-2020
Enter the Donation Amount:

250
Enter the Donation Type:

Fund
Enter the Donation Campaign Name:

Camp5
Enter  "1" if Org is Anonymous, "0"  if not:

1
Enter the Cheque Number:

567980
Query 9 suuccessfully executed
```

Run 4:

```
9
Enter values for Option 9

Enter Name of new Organization

BBC
Enter the mail address:

Jenkins Ave
Enter the Phone Number:

4056667771
Enter the Contact Person

Max
Organization inserted suuccessfully
Enter i : No of Donations, newly added Organization makes:

1
Enter "1" for Cheque and "2" for Creditcard donation

2
```


shehnazshaik@ou.edu

```
Enter the date of donation (DD-Mon-YYYY)::

20-Feb-2020
Enter the Donation Amount:

350
Enter the Donation Type:

Fund1
Enter the Donation Campaign Name:

Camp6
Enter  "1" if Org is Anonymous, "0"  if not:

0
Enter the Credit card Number:

1234734570
Enter the Card Type:

Visa
Enter the Card Expiry date (DD-Mon-YYYY)::

20-Jun-2023
Query 9 suuccessfully executed
```

Run 5:

```
9
Enter values for Option 9

Enter Name of new Organization

Org
Enter the mail address:

E Elhm
Enter the Phone Number:

8045556667
Enter the Contact Person

Harry
Organization inserted suuccessfully
Enter i : No of Donations, newly added Organization makes:

1
Enter "1" for Cheque and "2" for Creditcard donation

1

1
Enter the date of donation (DD-Mon-YYYY)::

05-Jan-2020
Enter the Donation Amount:

550
Enter the Donation Type:

Personal
Enter the Donation Campaign Name:

camp7
Enter  "1" if Org is Anonymous, "0"  if not:

1
Enter the Cheque Number:

120345
Query 9 suuccessfully executed
```

Records reflecting in DB:

shehnazshaik@ou.edu

```
SELECT * FROM Ext_Organization;
SELECT * FROM O_Cheque;
```

Results Messages

	O_Name	O_mail_Address	O_Phone_Num	O_ContactPerson
1	ABC	W Memorial	4056667771	Pratt
2	BBC	Jenkins Ave	4056667771	Max
3	BPF	SanDiego	8095741110	Chris
4	Carmax	Richmond	8095741111	Steve
5	CVS	W Main	4051112224	Sam
6	Lindt	E Brooks	4051112224	Rafal
7	Org	E Elhm	8045556667	Harry
8	Pepsico	W Elhm	4053334445	Chris
9	Philips	Houston	8095740001	Jim
10	TMobile	Vegas	8095749001	Jeff

```
SELECT * FROM Ext_Organization;
SELECT * FROM O_Cheque;
SELECT * FROM O_CreditCard;
```

Results Messages

	O_Name	O_DDate	O_DAmount	O_DType	O_DCamp_Name	O_isAnonymous	O_ChequeNum
1	ABC	2020-01-04	250	Fund	Camp5	1	567980
2	CVS	2020-01-01	550	Funds	Camp1	0	101120
3	Org	2020-01-05	550	Personal	camp7	1	120345
4	TMobile	2019-02-28	2500	Profits	OC2	0	123002

	O_Name	O_DDate	O_DAmount	O_DType	O_DCamp_Name	O_isAnonymous	O_CardNum	O_CardType	O_ExpDate
1	BBC	2020-02-20	350	Fund1	Camp6	0	1234734570	Visa	2023-06-20
2	BPF	2019-02-28	1800	Profits	OC2	0	9723899892	Discover	2022-01-31
3	CVS	2020-01-02	550	Compfund	Camp2	0	1234121310	Visa	2022-06-23
4	Pepsico	2020-01-03	1000	CompF	Camp3	0	123455431	Discover	2023-06-25
5	Philips	2019-01-31	2000	Crowd Fund	OC1	0	9723899891	Visa	2021-01-31

Task 6. 10 Testing query 10

Option 10 Query Execution:

Run 1:

```
10
Enter SSN of a Client

4513002
Retrieving Name and Ph. Number of Doctor for a particular team
Doctor_Name | Doctor_Phone
-----------------------------
Jarvis   | 8095789567
Query 10 suuccessfully executed
```


shehnazshaik@ou.edu

```
1   Select Doc_Name, Doc_Number from Client Where SSN = 4513002;
2
```

Results Messages

	Doc_Name	Doc_Number
1	Jarvis	8095789567

Run 2:

```
10
Enter SSN of a Client

4513001
Retrieving Name and Ph. Number of Doctor for a particular team
Doctor_Name | Doctor_Phone
-----------------------------
Karlo  | 4051112221
Query 10 suuccessfully executed
```

```
1   Select Doc_Name, Doc_Number from Client Where SSN = 4513001;
2
```

Results Messages

	Doc_Name	Doc_Number
1	Karlo	4051112221

6. 11 Testing query 11

Option 11 Query Execution:

Run 1:

```
11
Enter the Start Date (DD-Mon-YYYY):

01-Mar-2019
Enter the End Date (DD-Mon-YYYY):

01-Aug-2019
Emp_SSN  | Total_Expense
---------------------------------
4513003 | 150.0
4513007 | 1000.0
4513009 | 2500.0
```

```
1   Select SSN, SUM(E_Amount) from Expenses
2       Where E_Date BETWEEN '01-Mar-2019' AND '01-Aug-2019'
3       GROUP BY SSN ORDER BY SUM(E_Amount);
4
```

Results Messages

	SSN	(No column name)
1	4513003	150
2	4513007	1000
3	4513009	2500

Run 2:

shehnazshaik@ou.edu

```
11
Enter the Start Date (DD-Mon-YYYY):

01-Jun-2019
Enter the End Date (DD-Mon-YYYY):

31-Dec-2020
Emp_SSN   | Total_Expense
---------------------------------
4513006 | 300.0
4513011 | 500.0
4513007 | 1000.0
4513009 | 2500.0
```

```
50
51  Select SSN, SUM(E_Amount) from Expenses
52      Where E_Date BETWEEN '01-Jun-2019' AND '31-Dec-2019'
53      GROUP BY SSN ORDER BY SUM(E_Amount);
54
```

Results Messages

	SSN	(No column name)
1	4513006	300
2	4513011	500
3	4513007	1000
4	4513009	2500

6. 12 Testing query 12

Option 12 Query Execution:

```
12
Retrieving All the Client SSN to ultimately retrieve volunteers that are members of teams

4513001
4513002
4513005
4513007
4513012
Enter any client SSN from above results:

4513002
Retrieving Volunteers:

SSN       |       Name
--------------
4513005|Scott
4513007|Chang
4513011|Nina
Query 12 suuccessfully executed
```

```
32  SELECT SSN, Name FROM Person
33      WHERE SSN in (SELECT SSN FROM Serve WHERE T_name in
34      (SELECT T_name FROM Cares WHERE SSN='4513002'));
35
```

Results Messages

	SSN	Name
1	4513005	Scott
2	4513007	Chang
3	4513011	Nina

Run 2:

shehnazshaik@ou.edu

```
12
Retrieving All the Client SSN to ultimately retrieve volunteers that are members of teams

4513001
4513002
4513005
4513007
4513012
Enter any client SSN from above results:

4513005
Retrieving Volunteers:

SSN      |      Name
--------------
4513002|Stark
4513011|Nina
Query 12 suuccessfully executed
```

```
32    SELECT SSN, Name FROM Person
33    ····WHERE SSN in (SELECT SSN FROM Serve WHERE T_name in
34    ····(SELECT T_name FROM Cares WHERE SSN='4513005'));
35
```

Results Messages

	SSN	Name
1	4513002	Stark
2	4513011	Nina

6. 13 Testing query 13

Option 13 Query Execution:

```
13
Name    | Address       | email          | home_Num      | work_Num      | cell_Num
-------------------------------------------------------------------------------
Shanu |NORMAN    |Shane03@gmail.com |4058860000    |4058870001    |4058880002
Chang |Tulsa     |Chang3@gmail.com |4058860018    |4058870019    |4058880112
Jordan |Austin   |Jordan3@gmail.com |4058860800    |4058134000    |4058867500
```

```
36    SELECT Name, Address, email, home_Num, work_Num, cell_Num FROM Person WHERE SSN in
37    ····(SELECT SSN FROM Cares WHERE T_Name in
38    ····(SELECT T_Name FROM Sponsor WHERE O_Name BETWEEN 'B%' AND 'K%'));
39
```

Results Messages

	Name	Address	email	home_Num	work_Num	cell_Num
1	Shanu	NORMAN	Shane03@gmail.com	4058860000	4058870001	4058880002
2	Chang	Tulsa	Chang3@gmail.com	4058860018	4058870019	4058880112
3	Jordan	Austin	Jordan3@gmail.com	4058860800	4058134000	4058867500

6. 14 Testing query 14

shehnazshaik@ou.edu

Option 14 Query Execution:

```
14
Retriving Donors: If the Donor is Anonymous you will see 'Anonymous' as his or her name

Name    | Tot_Amount    | isAnonymous   |
-----------------------------------------
Jordan  |320.0  |False
Logan   |770.0  |False
Query 14 suuccessfully executed
```

```
40  SELECT P.Name, sum(DO.D_Amount) tot_Amt, D.isAnonymous as DO_Amt
41  FROM Donor D, Employee E, Person P,
42  (SELECT SSN, D_Amount from CreditCard
43  UNION ALL
44  SELECT SSN, D_Amount from Cheque) as DO
45  WHERE P.SSN = D.SSN
46  AND E.SSN = D.SSN
47  AND DO.SSN = D.SSN
48  GROUP BY D.SSN, P.Name, D.isAnonymous
49  ORDER BY DO_Amt;
```

Results Messages

	Name	tot_Amt	DO_Amt
1	Jordan	320	0
2	Logan	770	1

6. 15 Testing query 15

Option 15 Query Execution:

```
15
Enter the Date (DD-Mon-YYYY):

01-Jan-2011
Retrieving Team Names formed after above date
Team_Name
-----------
Bears
RCB
Query 15 suuccessfully executed
```

```
43  Select T_Name from Teams WHERE T_FormDate > '01-Jan-2011';
```

Results Messages

	T_Name
1	Bears
2	RCB

shehnazshaik@ou.edu

6. 16 Testing query 16

Option 16 Query Execution:

Before running option 16 in Java.

```
32    SELECT SSN, Salary FROM Employee;
33
```

Results Messages

	SSN	Salary
1	4513003	73700
2	4513006	85000
3	4513007	73000
4	4513009	78000
5	4513011	80000
6	4513012	70000

```
16
Updated Employee Table
Retrieving the Employee SSN and salary to check if update has been made
Emp_SSN         | Salary        |
----------------------------
4513003     |81070.0
4513006     |85000.0
4513007     |73000.0
4513009     |78000.0
4513011     |80000.0
4513012     |70000.0
Query 16 suuccessfully executed
```

```
32    SELECT SSN, Salary FROM Employee;
```

Results Messages

	SSN	Salary
1	4513003	81070
2	4513006	85000
3	4513007	73000
4	4513009	78000
5	4513011	80000
6	4513012	70000

shehnazshaik@ou.edu

6. 17 Testing query 17

Option 17 Query Execution:

Before running option 17 records in DB:

```
34  SELECT SSN FROM Client;
```

Results | Messages

	SSN
1	4513001
2	4513002
3	4513005
4	4513007
5	4513012

```
17
Record(s) Deleted
Checking if the records are deleted
Client_SSN
------------------
4513001
4513005
4513007
4513012
Query 17 suuccessfully executed
```

Client with SSN “4513002” has been deleted.

```
35  SELECT SSN FROM Client;
```

Results | Messages

	SSN
1	4513001
2	4513005
3	4513007
4	4513012

6. 18 Testing query 18

Option 18 Query Execution:

```
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
18
Enter the file name:
teamfile.txt
File has been successfuly Imported!
```


shehnazshaik@ou.edu

```
12    SELECT * FROM Teams;
```

Results | Messages

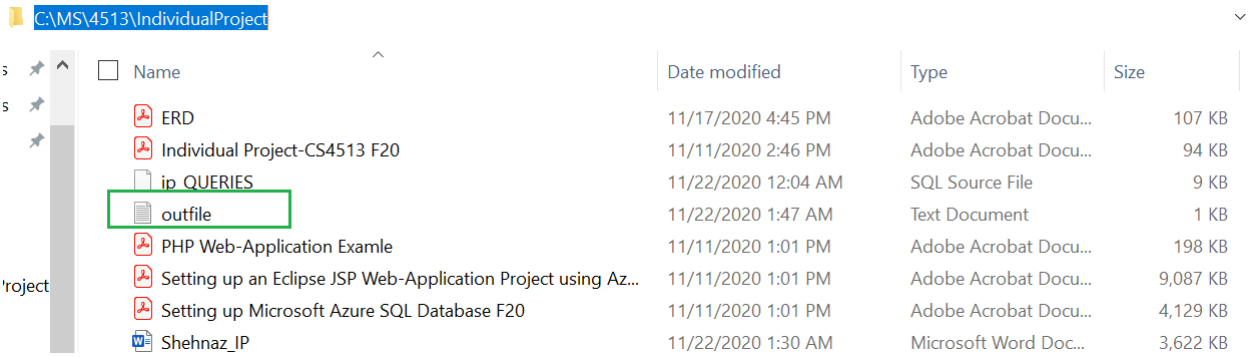
	T_Name	T_Type	T_FormDate
1	Bears	NFL	2012-02-15
2	CSK	IPL	2008-05-30
3	OKC Thunder	NBA	2010-12-14
4	OU	NCAA	2010-12-14
5	Rangers	Heroish	2013-03-13
6	RCB	IPL	2013-04-01
7	Riders	Race	2012-01-12
8	SRH	IPL	2008-08-25

6. 19 Testing query 19

Option 19 Query Execution:

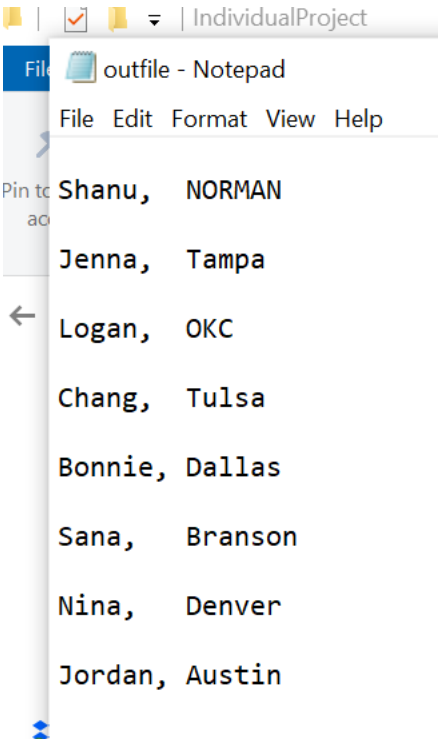
```
18
Enter the file name:
teamfile.txt
File has been successfuly Imported!

19
Enter the name of the file to be exported

outfile.txt
export successful
```

Outfile.txt has been Successfully created in the specified path in local.



Content of the exported file is the records of the retrieval query from Person table.

shehnazshaik@ou.edu



6. 20 Testing query 20

Option 20 Query Execution:

```
18
Enter the file name:
teamfile.txt
File has been successfuly Imported!

19
Enter the name of the file to be exported

outfile.txt
export successful

20
Quit.
 Exiting...
 Goodbye!
```

Errors:

Invalid Date: 30-02-2019

shehnazshaik@ou.edu

```
1
Enter the Team Name:

Team1
Enter the Team Type:

Race
Enter the Team Form date (DD-Mon-YYYY):

30-02-2019
We have an error: Conversion failed when converting date and/or time from character string.
```

Foreign Key Constraint error.

```
2
Enter values for Option 2

Enter SSN of new Client

10010101
Enter the Doctor Name:

testdoc
Enter the Attorney Name:

testatt
Enter the Doctor Number:

9990008887
Enter the Attorney Number:

9990008888
Enter the Organization assigned date (DD-Mon-YYYY):

09-Sep-2019
We have an error: The INSERT statement conflicted with the FOREIGN KEY constraint "FK__Client__SSN__436BFEE3". The conflict occurred in database "cs-dsa-4513-sql-db"
```

Data size for phone number was CHAR(10), Hence the error for giving 11 digits input.

```
2
Enter values for Option 2

Enter SSN of new Client

4513003
Enter the Doctor Name:

docnam
Enter the Attorney Name:

attnam
Enter the Doctor Number:

10102345671
Enter the Attorney Number:

2345556667
Enter the Organization assigned date (DD-Mon-YYYY):

01-Jan-2020
We have an error: String or binary data would be truncated in table 'cs-dsa-4513-sql-db.dbo.Client', column 'Doc_Number'. Truncated value: '1010234567'.
```

Task 7. JSP Web Application Run

7. 1 Source Program and compilation screenshots

Data Handler:

```
package jsp_azure_test;


import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;
```


shehnazshaik@ou.edu

```
import java.sql.DriverManager;
import java.sql.PreparedStatement;


public class DataHandler {


   private Connection conn;


   // Azure SQL connection credentials
   private String server = "shai0002.database.windows.net";
   private String database = "cs-dsa-4513-sql-db";
   private String username = "shai0002";
   private String password = "$Shhane031";


   // Resulting connection string
   final private String url =

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


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


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


shehnazshaik@ou.edu

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

    final String sqlQuery = "SELECT T_Name FROM Teams WHERE T_FormDate > '" + form_date + "';";
    final PreparedStatement stmt = conn.prepareStatement(sqlQuery);
    return stmt.executeQuery();
}

// Inserts a record into the movie_night table with the given attribute values
public boolean addTeam(
        String T_name, String T_type, String formDate) throws SQLException {

    getDBConnection(); // Prepare the database connection

    // Prepare the SQL statement
    final String sqlQuery =
            "INSERT INTO Teams VALUES(?, ?, ?)";

    final PreparedStatement stmt = conn.prepareStatement(sqlQuery);

    // Replace the '?' in the above statement with the given attribute values
    stmt.setString(1, T_name);
    stmt.setString(2, T_type);
    stmt.setString(3, formDate);


    // Execute the query, if only one record is updated, then we indicate success by returning true
    return stmt.executeUpdate() == 1;
```


shehnazshaik@ou.edu

```
  }
}
```

Add_team_form.jsp

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

Add_team.jsp

shehnazshaik@ou.edu

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

    // Get the attribute values passed from the input form.
    String T_name = request.getParameter("team_name");
    String T_type = request.getParameter("team_type");
    String form_date = request.getParameter("form_date");
    /*
     * If the user hasn't filled out all the time, movie name and duration. This is
very simple checking.
     */
    if (T_name.equals("") || T_type.equals("") || form_date.equals("")) {
        response.sendRedirect("add_team_form.jsp");
    } else {

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

            <h2>Was successfully inserted.</h2>
            <%
        }
    }
    %>
    </body>
</html>
```

enterDate.jsp:

```
<!DOCTYPE html>
<html>
```


shehnazshaik@ou.edu

```
    <head>
        <meta charset="UTF-8">
        <title>Get Teams formed after a Date</title>
    </head>
    <body>
        <h2>Get Teams formed after a Date</h2>
        <!--
            Form for collecting user input for the new movie_night record.
            Upon form submission, add_movie.jsp file will be invoked.
        -->
        <form action="get_all_teams.jsp">
            <!-- The form organized in an HTML table for better clarity. -->
            <table border=1>
                <tr>
                    <th colspan="2">Enter the Formation Team Date:</th>
                </tr>

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

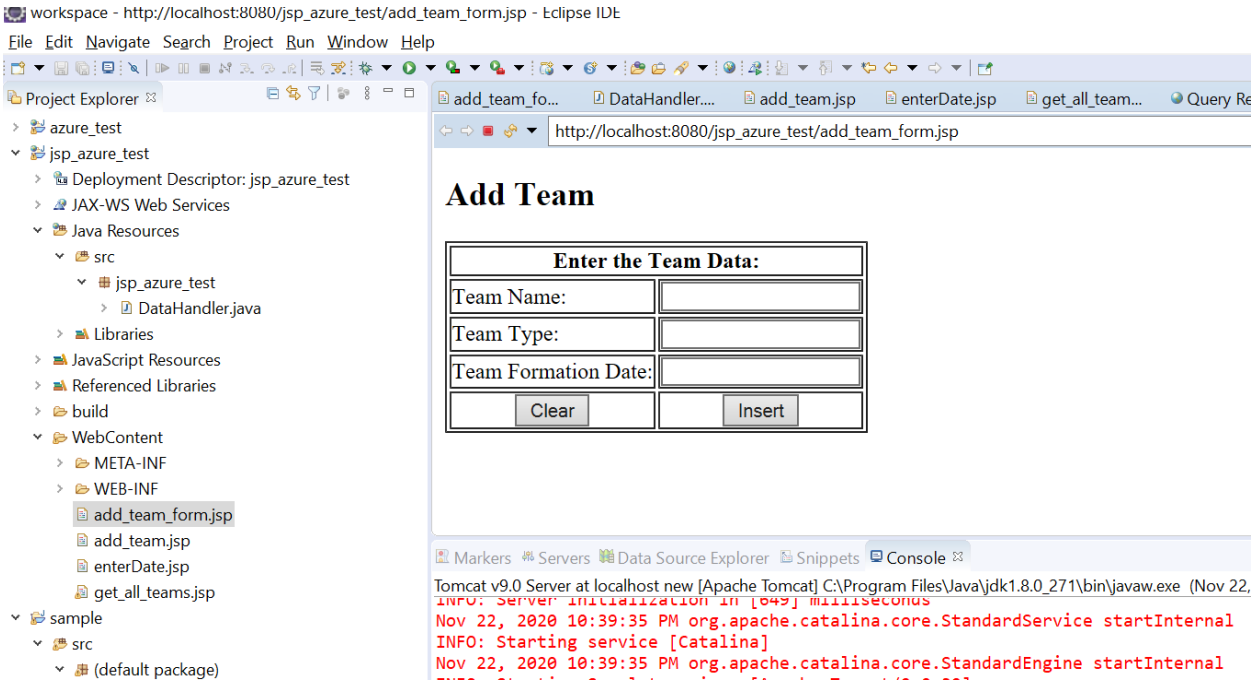
Get_all_teams.jsp:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
    <meta charset="UTF-8">
        <title>Movie Nights</title>
    </head>
    <body>
        <%@page import="jsp_azure_test.DataHandler"%>
        <%@page import="java.sql.ResultSet"%>
        <%
            // We instantiate the data handler here, and get all the movies from the
database
            final DataHandler handler = new DataHandler();
            String form_date = request.getParameter("form_date");
            final ResultSet teams = handler.getAllTeams(form_date);
        %>
```


shehnazshaik@ou.edu

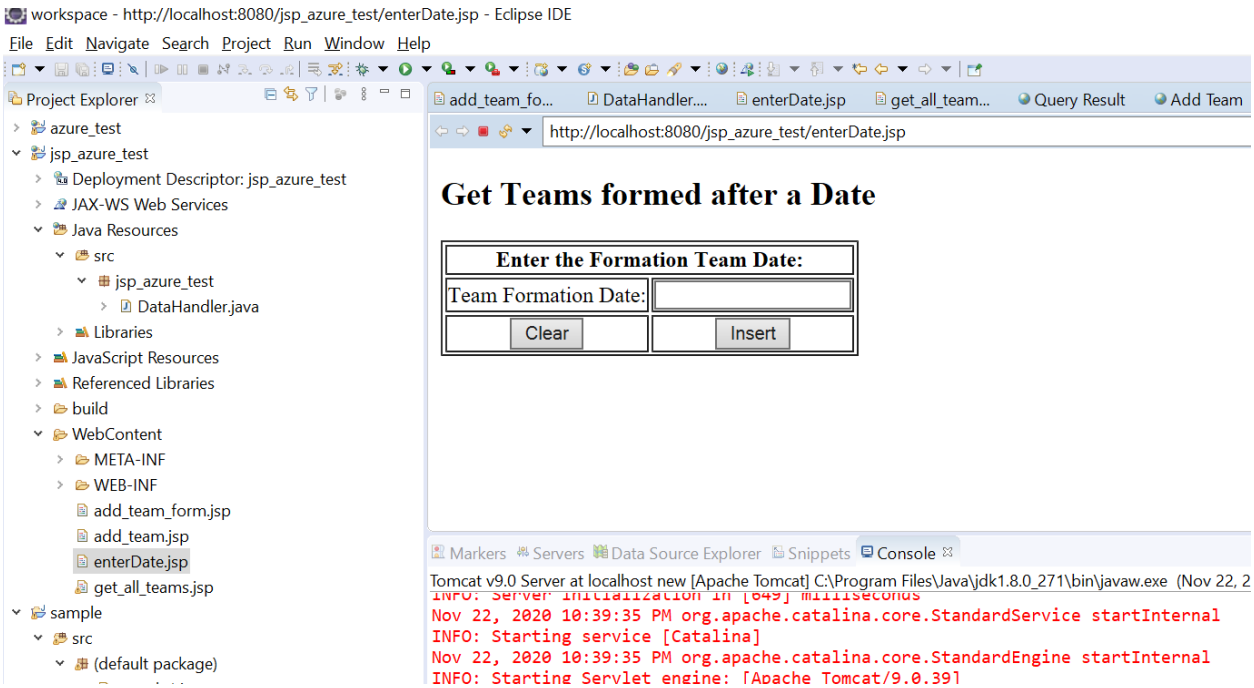
```
        <!-- The table for displaying all the movie records -->
        <table cellspacing="2" cellpadding="2" border="1">
            <tr> <!-- The table headers row -->
              <td align="center">
                <h4>Team Name</h4>
              </td>
            </tr>
            <%
               while(teams.next()) { // For each movie_night record returned...
                   // Extract the attribute values for every row returned
                   final String T_name = teams.getString(1);

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

Add Team Form compilation (For Query 1):



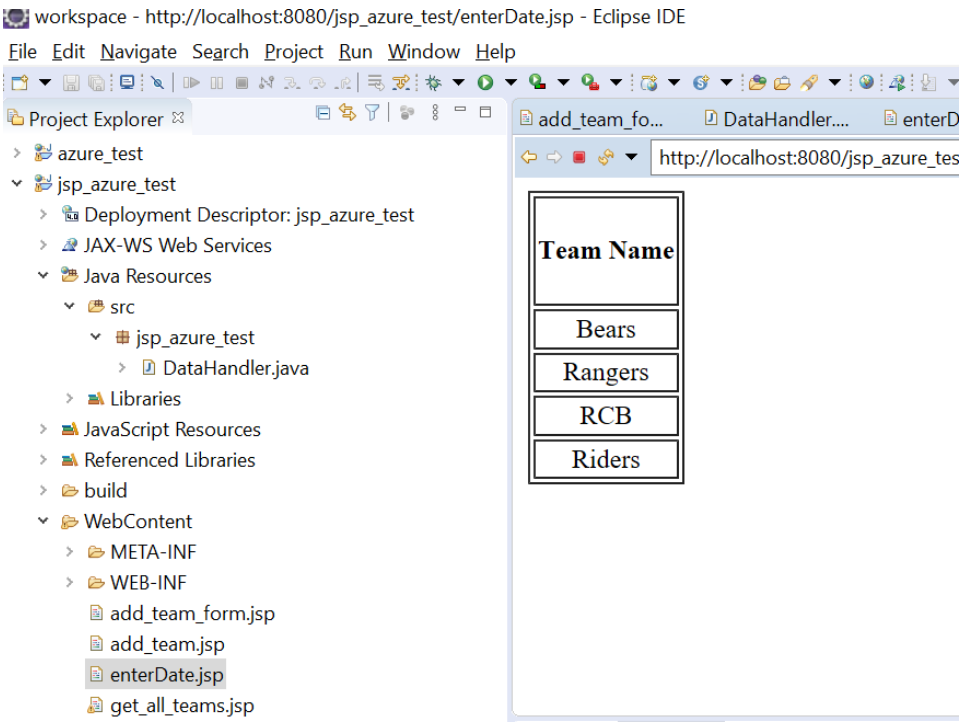
Enter Date (For Query 15) Compilation:

shehnazshaik@ou.edu

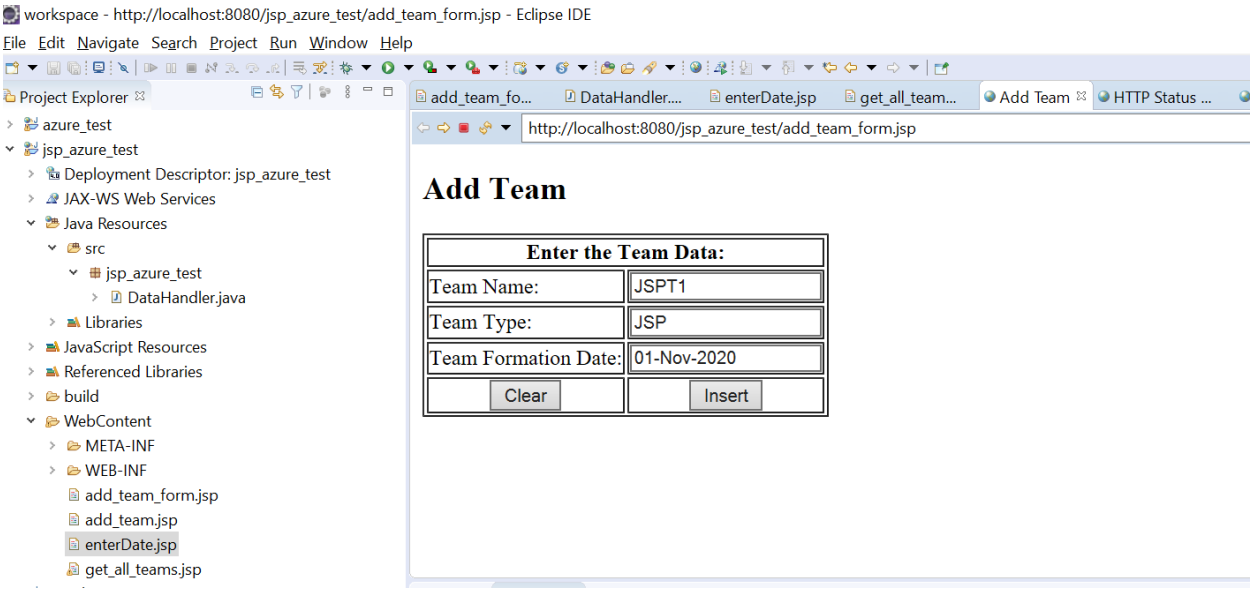


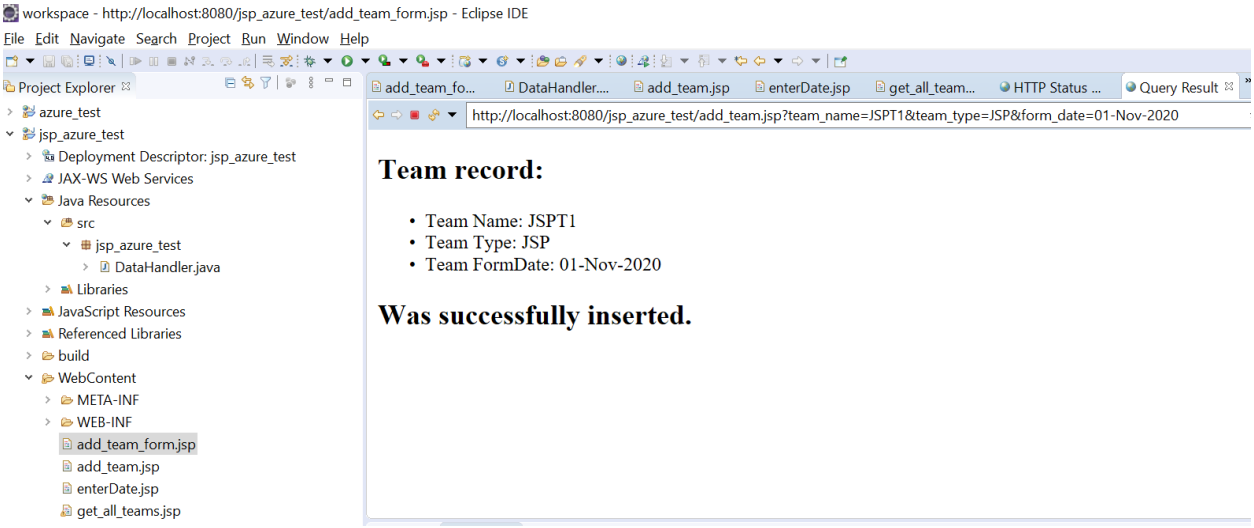
7. 2 Testing query 2

Query 15:

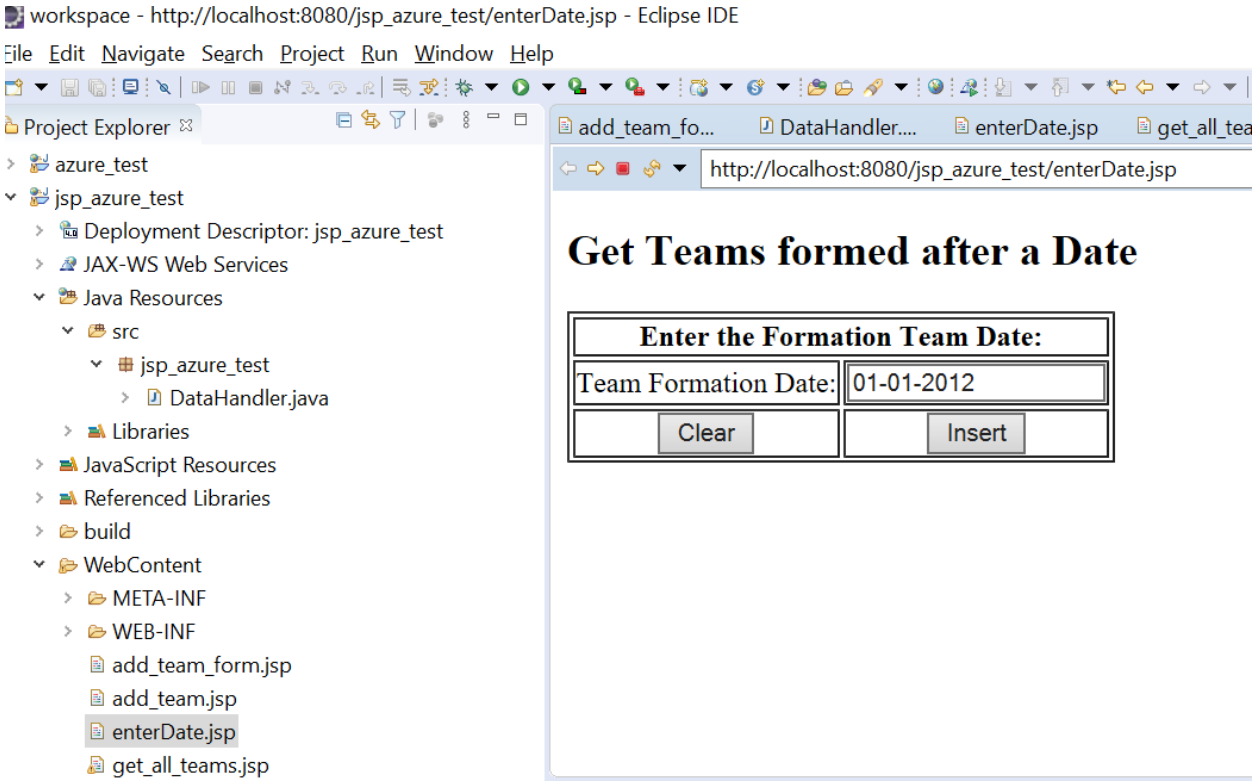


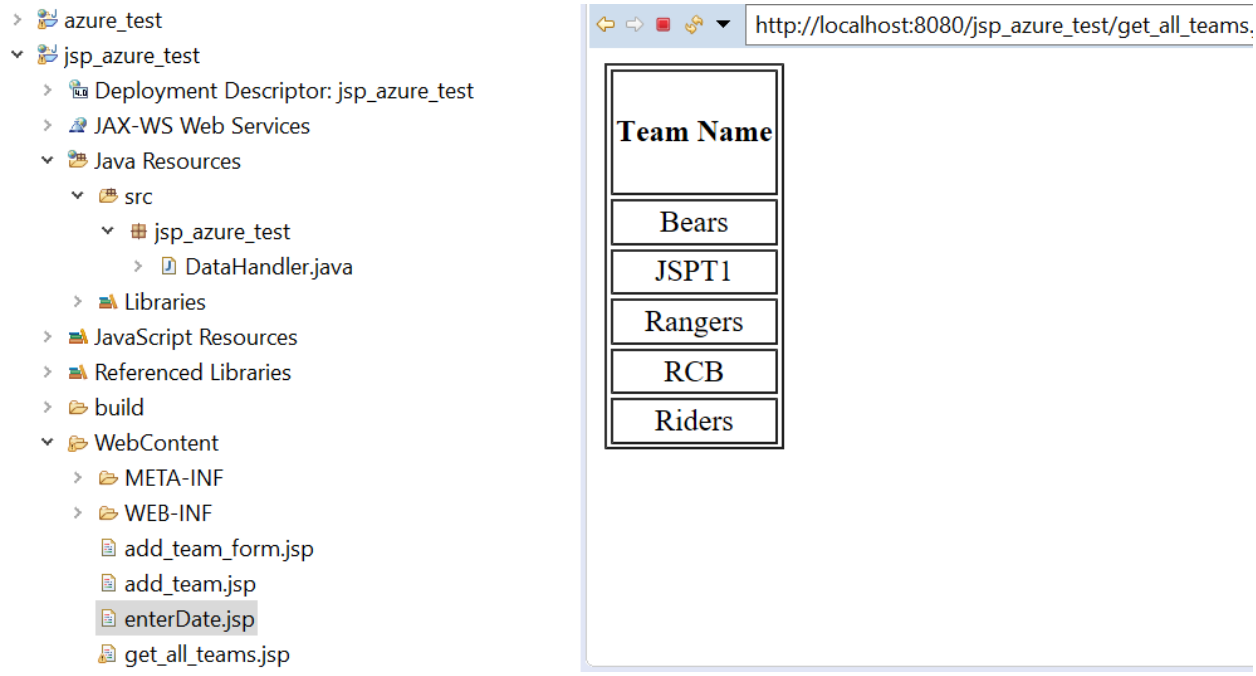
shehnazshaik@ou.edu





shehnazshaik@ou.edu





shehnazshaik@ou.edu