Cover Page

Course: CS/DSA-4513 – Database Management

Section: 001

Semester: FALL 2024

Instructor: DR. LE GRUENWALD

Group Number: 36

Student: Kazi Priom

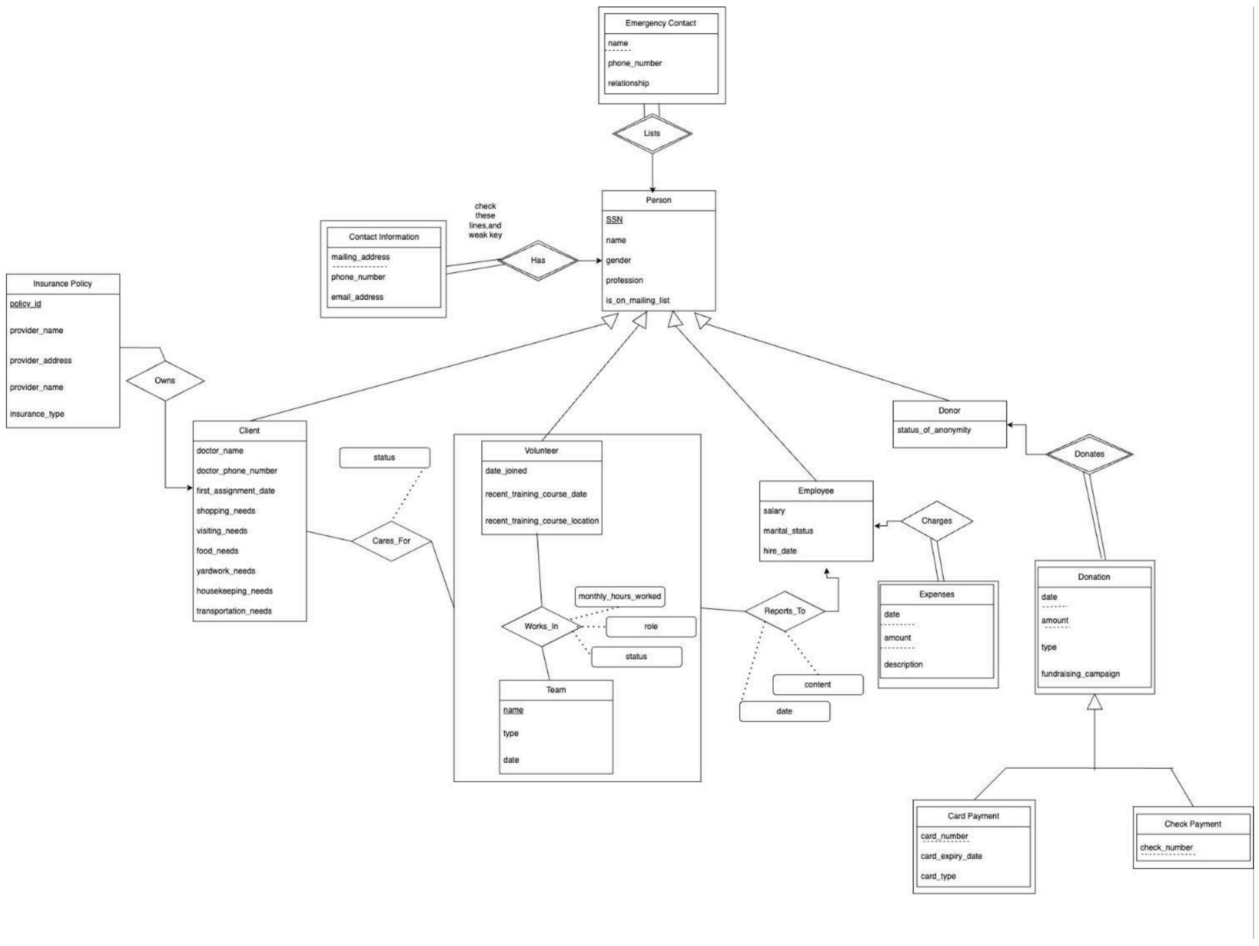
ID: 113599594

Email: kazi.s.priom-1@ou.edu

Individual Project

Score:

Task 1 ER Diagram 1



Task 2: Relational Database Schemas

Person(SSN, name, gender, profession, is_on_mailing_list)

Emergency_Contact(SSN, name, phone_number, relationship)
Contact_Information(SSN, mailing_address, phone_number, email_address)

Client(SSN, doctor_name, doctor_phone_number, first_assignment_date, shopping_needs, visiting_needs, food_needs, yardwork_needs, housekeeping_needs, transportation_needs)

Insurance_Policy(policy_id, provider_name, provider_address, provider_name, insurance_type)

Owns(policy_id, SSN)

Volunteer(SSN, date_joined, recent_training_course_date, recent_training_course_location)

Team(name, type, date)

Works_In(SSN, name, monthly_hours_worked, role, status)

Reports_To(volunteer_SSN, name, monthly_hours_worked, date, content, employee_SSN)

Cares_For(volunteer_SSN, name, monthly_hours_worked, status, client_SSN)

Employee (SSN, salary, marital_status, hire_date)

Expenses (SSN, date, amount, description)

Donor(SSN, status_of_anonymity)

Donation(SSN, date, amount, type, fundraising_campaign)

Card_Payment(SSN, date, amount, card_number, card_expiry_date, card_type)

Check_Payment(SSN, date, amount, check_number)

Task 3: Discussion of storage structures for tables 12-14

3.2. Discussion of storage structures for tables (Azure SQL Database) 15-20

Table Name	Query # and Type	Search Key	Query Frequency	Selected File organization	Justification

Person	2. Insertion 3. Insertion 4. Random Search 5. Insertion 6. Random Search 7. Insertion 8. Random Search 9. Random Search 10. Random Search 12. Range Search 13. Random Search 14. Random search 15. Deletion	4. SSN 6. SSN 8. SSN 9. SSN 10.SSN, name 12. SSN, name 13. SSN, name 14. SSN 15. SSN	2. 1/week 3. 2/month 4. 30/month 5. 1/year 6. 1/day 7. 1/day 8. 1/week 9. 1/month 10. 4/year 12. 1/week 13. 1/week 14. 1/year 15. 4/year	**B+ Tree**	A B+ tree is great because of the sheer amount of insertions, deletions, and searches. Only a tree can provide such versatility.
Emergency_Contact	12. Range Search	12. SSN, name, phone_number, relationship	12. 1/week	**Dense Sequential File**	We care about fast searching because that is all we do for this table. No insertion or extra space needed which would remove B+ tree overhead. Also range searches are not good on hash indexes.
Contact_Information	12. Range Search	12. SSN, mailing_address, phone_number, email_address	12. 1/week	**Dense Sequential File**	We care about fast searching because that is all we do for this table. No insertion or extra space needed which would remove B+ tree overhead. Also range searches are not good on hash indexes.

Client	2. Insertion 8. Random Search 10. Random Search 12. Range Search 15. Deletion, Range Search	8. SSN, doctor_name, doctor_phone_number 10. SSN 12. SSN, name 15. transportation_needs	2. 1/week 8. 1/week 10. 4/year 12. 1/week 15. 4/year	**B+ Tree**	A B+ tree is great because of the frequent insertions, deletions, and searches. Only a tree can provide such versatility.
Insurance_Policy	15. Random Search	15. insurance_type	15. 4/year	**Hash Index**	Nothing beats O(1) time especially when all we worry about is random searching.
Owns	15. Random Search, Deletion	15. SSN	15. 4/year	**Hash index**	O(1) time in random searching and deletion.
Volunteer	3. Insertion 10. Random Search 12. Range Search	10. SSN, name 12. SSN, name	3. 2/month 10. 4/year 12. 1/week	**Dense Sequential File**	A B+ tree is usually good here, but the queries are infrequent, so we can use a simpler file organization.
Team	1. Insertion 11. Range Search	11. Name, date	1. 1/month 11. 1/month	**Dense Sequential File**	A B+ tree is usually good here, but the queries are infrequent, so we can use a simpler file organization.
Works_In	(given team name and volunteer SSN is known) 3. Insertion 4. Insertion		3. 2/month 4. 30/month	**Heap Files**	Easy and quick way to deal with insertions as that's all that is needed here.
Reports_To	5. Insertion 14. Range Search	14. employee_SSN	5. 1/year 14. 1/year	**Dense Sequential File**	A B+ tree is usually good here, but the queries

					are infrequent, so we can use a simpler file organization.
Cares_For	(given volunteer_SSN and team name are known) 2. Insertion 10. Random Search 15. Deletion	10. volunteer_SSN, client_SSN, status	2. 1/week 10. 4/year 15. 4/year	**Heap Files**	Easy and quick way to deal with frequent insertions. Queries 10 and 15 don't happen often, so B+ Tree is not worth it.
Employee	5. Insertion 12. Range Search 13. Random Search 14. Range Search	12. SSN, name 13. SSN, name 14. SSN, salary	12. 1/week 13. 1/week 14. 1/year	**B+ Tree**	Overall best with insertion and range search.
Expenses	6. Insertion 7. Range Search	7. SSN, date, amount	6. 1/day 7. 1/day	**B+ Tree**	Frequent insertions and range searches require a B+ Tree.
Donor	7. Insertion 13. Random Search	13. SSN, status_of_anonymity, name	7. 1/day 13. 1/week	**B+ Tree**	Frequent insertions and range searches require a B+ Tree.
Donation	13. Random Search	13. SSN, date, amount	13. 1/week	**Hash Index**	Nothing beats O(1) time especially when all we worry about is random searching.
Card_Payment					
Check_Payment					

Task
3.2. Discussion of storage structures for tables (Azure SQL Database)
15-20

Now we have new structures in Azure including :

clustered index: good for range-searches and random searches. Changes the physical data by sorting the information. This is usually done on primary keys, but weak entity sets can do it on foreign keys. The tables that use this structure benefit heavily from clustering
Tables: Person, Client, Volunteer, Employee, Expenses, Donor

non-clustered index: good for random searches without changing the physical data.
Tables: Emergency_Contact, Contact_Information, Reports_To, Volunteer, Owns, Donation

None: there is no need to change some tables as they are more efficient by their relations
Cares_For
Card_Payment
Check_Payment

Task 4: SQL statements and screenshots showing the creation of 21-50 tables in Azure SQL Database

```
drop procedure if exists AddNewDonor;
drop table if exists Check_Payment;
drop table if exists Card_Payment;
drop type if exists DonType;
drop table if exists Donation;
drop table if exists Expenses;
drop table if exists Cares_For;
drop table if exists Reports_To;
drop table if exists Works_In;
drop table if exists Team;
drop table if exists Volunteer;
drop table if exists Owns;
drop table if exists Insurance_Policy;
drop table if exists Client;
drop table if exists Contact_Information;
drop table if exists Emergency_Contact;
drop table if exists Donor;
drop table if exists Employee;
drop table if exists Person;



declare @i int = 0
```

```
create table Person (
   SSN int primary key clustered,
   name nvarchar(64),
   gender nvarchar(10),
   profession nvarchar(64),
   is_on_mailing_list bit
);



create table Emergency_Contact (
   SSN int,
   name nvarchar(64),
   phone_number nvarchar(15),
   relationship nvarchar(32),
   foreign key (SSN) references Person(SSN)
);
create nonclustered index idx_EC_SSN on Emergency_Contact(SSN);

create table Contact_Information (
   SSN int,
   mailing_address nvarchar(255),
   phone_number nvarchar(15),
   email_address nvarchar(64),
   foreign key (SSN) references Person(SSN)
);
create nonclustered index idx_CI_SSN on Contact_Information(SSN);


create table Client (
   client_SSN int primary key clustered,
   doctor_name nvarchar(64),
   doctor_phone_number nvarchar(15),
   first_assignment_date date,
   shopping_needs int check (shopping_needs between 1 and 10),
   visiting_needs int check (visiting_needs between 1 and 10),
   food_needs int check (food_needs between 1 and 10),
   yardwork_needs int check (yardwork_needs between 1 and 10),
   housekeeping_needs int check (housekeeping_needs between 1 and 10),
   transportation_needs int check (transportation_needs between 1 and 10),
   foreign key (client_SSN) references Person(SSN)
);
```

```
create table Insurance_Policy (
   policy_id int primary key nonclustered,
   provider_name nvarchar(64),
   provider_address nvarchar(255),
   insurance_type nvarchar(64)
);

create nonclustered index IX_Insurance_Type on Insurance_Policy(insurance_type);

create table Owns (
   policy_id int,
   client_SSN int,
   foreign key (policy_id) references Insurance_Policy(policy_id),
   foreign key (client_SSN) references Client(client_SSN)
);
create nonclustered index idx_Owns_SSN on Owns(client_SSN);

create table Volunteer (
   vol_SSN int primary key clustered,
   date_joined date,
   recent_training_course_date date,
   recent_training_course_location nvarchar(64),
   foreign key (vol_SSN) references Person(SSN)
);
create table Team (
   team_name nvarchar(64),
   team_type nvarchar(64),
   date date,
   primary key (team_name)
);
create table Employee (
   emp_SSN int primary key clustered,
   salary decimal(10, 2),
   marital_status nvarchar(16),
   hire_date date,
   foreign key (emp_SSN) references Person(SSN)
);
create table Works_In (
   SSN int,
   name nvarchar(64),
   monthly_hours_worked int,
   role nvarchar(64),
   status nvarchar(32),
   foreign key (SSN) references Volunteer(vol_SSN),
   foreign key (name) references Team(team_name)
);


create table Reports_To (
   volunteer_SSN int,
   team_name nvarchar(64),
   date date,
   content nvarchar(255),
   employee_SSN int,
   foreign key (volunteer_SSN) references Volunteer(vol_SSN),
   foreign key (employee_SSN) references Employee(emp_SSN),
   constraint reportingTeam foreign key (team_name) references Team(team_name)
);
create nonclustered index idx_RT_volunteer_SSN on Reports_To(volunteer_SSN);


create table Cares_For (
   volunteer_SSN int,
   name nvarchar(64),
   status bit,
   client_SSN int,
   foreign key (volunteer_SSN) references Volunteer(vol_SSN),
   foreign key (client_SSN) references Client(client_SSN)
);




create table Expenses (
   SSN int,
   date date,
   amount decimal(10, 2),
   description nvarchar(255),
   foreign key (SSN) references Employee(emp_SSN)
);
create clustered index idx_Expenses_SSN on Expenses(SSN);
create table Donor (
   donr_SSN int primary key clustered,
   status_of_anonymity bit,
   foreign key (donr_SSN) references Person(SSN)
);

create table Donation (
   SSN int,
   date date,
   amount decimal(10, 2),
   type nvarchar(32),
   fundraising_campaign nvarchar(64),
   primary key nonclustered (SSN, date)
);

create type DonType AS TABLE
(
   date date,
   amount decimal(10, 2),
   type nvarchar(32),
   fundraising_campaign nvarchar(64)
);

create table Card_Payment (
   SSN int,
   date date,
   amount decimal(10, 2),
   card_number nvarchar(16),
   card_expiry_date date,
   card_type nvarchar(16),
   foreign key (SSN, date) references Donation(SSN, date)
);
create nonclustered index idx_cp_SSN on Card_Payment(SSN);

create table Check_Payment (
   SSN int,
   date date,
   amount decimal(10, 2),
   check_number nvarchar(16),
   foreign key (SSN, date) references Donation(SSN, date)
);
create nonclustered index idx_ChP_SSN on Check_Payment(SSN);
```

```
--basic proced.
go
drop procedure if exists AddEmergencyContacts
go

create procedure AddEmergencyContacts(
    @SSN int,
    @name nvarchar(64),
    @phone_number nvarchar(15),
    @relationship nvarchar(32)
)
as
begin
    insert into Emergency_Contact(SSN, name, phone_number, relationship)
    values(@SSN, @name, @phone_number, @relationship)
end;

go
drop procedure if exists addContactInfo;
go
create procedure addContactInfo(
    @SSN int,
    @mailing_address nvarchar(255),
    @phone_number nvarchar(15),
    @email_address nvarchar(64)
)
as
begin
    insert into Contact_Information(SSN, mailing_address, phone_number, email_address)
    values(@SSN, @mailing_address, @phone_number, @email_address);
end;

go
drop procedure if exists ClientInsurance
go

create procedure ClientInsurance(
    @policy_id int,
    @provider_name nvarchar(64),
    @provider_address nvarchar(255),
    @insurance_type nvarchar(64)
)
as
begin
   insert into Insurance_Policy(policy_id, provider_name, provider_address, insurance_type)
   values(@policy_id, @provider_name, @provider_address, @insurance_type);
end;
go
drop procedure if exists AddDonation
go

create procedure AddDonation
   @SSN int,
   @date date,
   @amount decimal(10, 2),
   @type nvarchar(32),
   @fundraising_campaign nvarchar(64)



AS
BEGIN
   -- Insert a single donation into the Donations table
   INSERT INTO Donation (SSN, date, amount, type, fundraising_campaign)
   VALUES (@SSN, @date, @amount, @type, @fundraising_campaign);


END;
GO

GO
DROP PROCEDURE IF EXISTS AddToWorksIn;
GO

CREATE PROCEDURE AddToWorksIn
   @volSSN INT,
   @team_name NVARCHAR(64),
   @monthly_hours_worked INT,
   @role NVARCHAR(64),
   @status NVARCHAR(32)
AS
BEGIN
   -- Insert into Works_In table
```

```
    insert into Works_In (SSN, name, monthly_hours_worked, role, status)
    VALUES (@volSSN, @team_name, @monthly_hours_worked, @role, @status);

END;
GO



GO
DROP PROCEDURE IF EXISTS AddToReportsTo;
GO

CREATE PROCEDURE AddToReportsTo
    @volunteer_ssn INT,
    @team_name NVARCHAR(64),
    @reporting_date DATE,
    @content NVARCHAR(255),
    @employee_ssn INT
AS
BEGIN
    -- Insert into Reports_To table
    INSERT INTO Reports_To (volunteer_SSN, team_name, date, content, employee_SSN)
    VALUES (@volunteer_ssn, @team_name, @reporting_date, @content, @employee_ssn);

END;
GO






--query 1: make a new team
go
drop procedure if exists AddNewTeam;
GO

create procedure AddNewTeam(
    @team_name nvarchar(50),
    @team_type nvarchar(50),
    @date date
)
as
```

```
begin
   insert into Team (team_name, team_type, date)
   values (@team_name, @team_type, @date);
end
Go
--query 2: add a new client and associate with team
drop procedure if exists AddNewClient;
go
create procedure AddNewClient(
   --client info
   @client_SSN int,
   @name nvarchar(50),
   @gender nvarchar(10),
   @profession nvarchar(50),
   @is_on_mailing_list bit,
   @doctor_name nvarchar(50),
   @doctor_phone_number nvarchar(15),
   @first_assignment_date date,
   @shopping_needs int,
   @visiting_needs int,
   @food_needs int,
   @yardwork_needs int,
   @housekeeping_needs int,
   @transportation_needs int,

   --team info
   @volunteer_SSN int,
   @team_name nvarchar(64),
   @status bit,

   --emergency contact
   @emer_name nvarchar(64),
   @emer_phone_number nvarchar(15),
   @emer_relationship nvarchar(32),

   --contact info
   @client_mailing_address nvarchar(255),
   @client_phone_number nvarchar(15),
   @client_email_address nvarchar(64),

   --insurance policy
   @policy_id int,
   @provider_name nvarchar(64),
   @provider_address nvarchar(255),
   @insurance_type nvarchar(64)
)
as
begin
   insert into Person (SSN, name, gender, profession, is_on_mailing_list)
   values (@client_SSN, @name, @gender, @profession, @is_on_mailing_list);

   insert into Client (client_SSN, doctor_name, doctor_phone_number, first_assignment_date, shopping
   values (@client_SSN, @doctor_name, @doctor_phone_number, @first_assignment_date, @shopping_needs,

   insert into Cares_For(volunteer_SSN, name, status, client_SSN)
   values(@volunteer_SSN, @name, @status, @client_SSN);

   --emergency contact
   exec AddEmergencyContancts @client_SSN, @emer_name, @emer_phone_number, @emer_relationship;
   --contact info
   exec addContanctInfo @client_SSN,@client_mailing_address,
    @client_phone_number, @client_email_address

   --insurance policy
   exec ClientInsurance @policy_id, @provider_name, @provider_address,
   @insurance_type
   insert into Owns(policy_id, client_SSN)
   values(@policy_id, @client_SSN)

end


--query 3: add a new volunteer and associate with teams
go
drop procedure if exists AddNewVolunteer;
GO
create procedure AddNewVolunteer(
   --values to add new volunteer
   @volunteer_SSN int,
   @name nvarchar(50),
   @gender nvarchar(10),
   @profession nvarchar(50),
   @is_on_mailing_list bit,
   @date_joined date,
   @recent_training_course_date date,
   @recent_training_course_location nvarchar(50),
   -- information to associate with team
   @team_name nvarchar(50),

   --emergency contact
   @emer_name nvarchar(64),
   @emer_phone_number nvarchar(15),
   @emer_relationship nvarchar(32),

   --contact info
   @vol_mailing_address nvarchar(255),
   @vol_phone_number nvarchar(15),
   @vol_email_address nvarchar(64)
)
as
begin

       -- insert person
       insert into Person (SSN, name, gender, profession, is_on_mailing_list)
       values (@volunteer_SSN, @name, @gender, @profession, @is_on_mailing_list);

   insert into Volunteer (vol_SSN, date_joined, recent_training_course_date, recent_training_course_
   values (@volunteer_SSN, @date_joined, @recent_training_course_date, @recent_training_course_locat

   insert into Works_In (SSN, name, monthly_hours_worked, role, status)
   values (@volunteer_SSN, @team_name, 0, 'member', 'active');

   --emergency contanct
   exec AddEmergencyContancts @volunteer_SSN, @emer_name, @emer_phone_number, @emer_relationship;
   --contact info
   exec addContanctInfo @volunteer_SSN,@vol_mailing_address, @vol_phone_number,
    @vol_email_address;
end;

--query 4: update monthly hours worked on a team
go
drop procedure if exists UpdateVolunteerHours;
GO
create procedure UpdateVolunteerHours(
   @SSN int,
   @team_name nvarchar(50),
   @hours int
)
as
begin
   update Works_In
   set monthly_hours_worked = monthly_hours_worked + @hours
   where SSN = @SSN and name = @team_name;
end;

--query 5: Enter a New Employee and Associate with Teams
go
drop procedure if exists AddNewEmployee;
GO
create procedure AddNewEmployee(
   -- Data to add new employee
   @employee_SSN int,
   @name nvarchar(50),
   @gender nvarchar(10),
   @profession nvarchar(50),
   @is_on_mailing_list bit,
   @salary decimal(10, 2),
   @marital_status nvarchar(10),
   @hire_date date,
   -- Data to associate with team
   @volunteer_SSN int,
   @team_name nvarchar(50),
   @date date,
   @content nvarchar(255),
   --emergency contact
   @emer_name nvarchar(64),
   @emer_phone_number nvarchar(15),
   @emer_relationship nvarchar(32),

   --contact info
   @emp_mailing_address nvarchar(255),
   @emp_phone_number nvarchar(15),
   @emp_email_address nvarchar(64)
)
as
begin
   -- insert employee into Person table
```

```
        -- insert person
        insert into Person (SSN, name, gender, profession, is_on_mailing_list)
        values (@employee_SSN, @name, @gender, @profession, @is_on_mailing_list);

    -- employee table
    insert into Employee (emp_SSN, salary, marital_status, hire_date)
    values (@employee_SSN, @salary, @marital_status, @hire_date);


    --emergency contact
    exec AddEmergencyContancts @employee_SSN, @emer_name, @emer_phone_number, @emer_relationship;

    --contact info
    exec addContanctInfo @employee_SSN,@emp_mailing_address, @emp_phone_number,
    @emp_email_address;

    -- Step 5: Insert into Reports_To table
    insert into Reports_To (volunteer_SSN, team_name, date, content, employee_SSN)
    values (@volunteer_SSN, @team_name, @date, @content, @employee_SSN);
end;

--q6: add expense
go
drop procedure if exists AddExpense;
go
create procedure AddExpense(
    @SSN int,
    @date date,
    @amount decimal(10, 2),
    @description nvarchar(255)
)
as
begin
    insert into Expenses (SSN, date, amount, description)
    values (@SSN, @date, @amount, @description);
end;

--q7 add new donor
go
drop procedure if exists AddNewDonor;
go
create procedure AddNewDonor(
    --donor info
    @don_SSN int,
    @name nvarchar(50),
    @gender nvarchar(10),
    @profession nvarchar(50),
    @is_on_mailing_list bit,
    @status_of_anonymity bit,

    --emergency contact
    @emer_name nvarchar(64),
    @emer_phone_number nvarchar(15),
    @emer_relationship nvarchar(32),

    --contact info
    @don_mailing_address nvarchar(255),
    @don_phone_number nvarchar(15),
    @don_email_address nvarchar(64)
)
as
begin
    --new person if no SSN
    if not exists (select 1 from Person where SSN = @don_SSN)
    begin
        -- insert person
        insert into Person (SSN, name, gender, profession, is_on_mailing_list)
        values (@don_SSN, @name, @gender, @profession, @is_on_mailing_list);
    end;

    -- donor
    insert into Donor (donr_SSN, status_of_anonymity)
    values (@don_SSN, @status_of_anonymity);

   --emergency contact
    exec AddEmergencyContancts @don_SSN, @emer_name, @emer_phone_number, @emer_relationship;

    --contact info
    exec addContanctInfo @don_SSN,@don_mailing_address,@don_phone_number,@don_email_address;

END;



--q8 doctor info
```

```
go
drop procedure if exists GetClientDoctorInfo;
go
create procedure GetClientDoctorInfo(
   @SSN int
)
as
begin
   select doctor_name, doctor_phone_number
   from Client
   where client_SSN = @SSN;
end;

--q9 Retrieve Total Expenses Charged by Each Employee for a Period
go
drop procedure if exists GetEmployeeExpenses;
go
create procedure GetEmployeeExpenses(
   @start_date date,
   @end_date date
)
as
begin
   select e.emp_SSN, sum(exp.amount) as total_expenses
   from Employee e
   join Expenses exp on e.emp_SSN = exp.SSN
   where exp.date between @start_date and @end_date
   group by e.emp_SSN
   order by total_expenses desc;
end;

--q10 Retrieve the List of Volunteers Supporting a Particular Client
go
drop procedure if exists GetVolunteersSupportingClient;
go
create procedure GetVolunteersSupportingClient(
   @client_SSN int
)
as
begin
   select distinct p.name
   from Person p
```

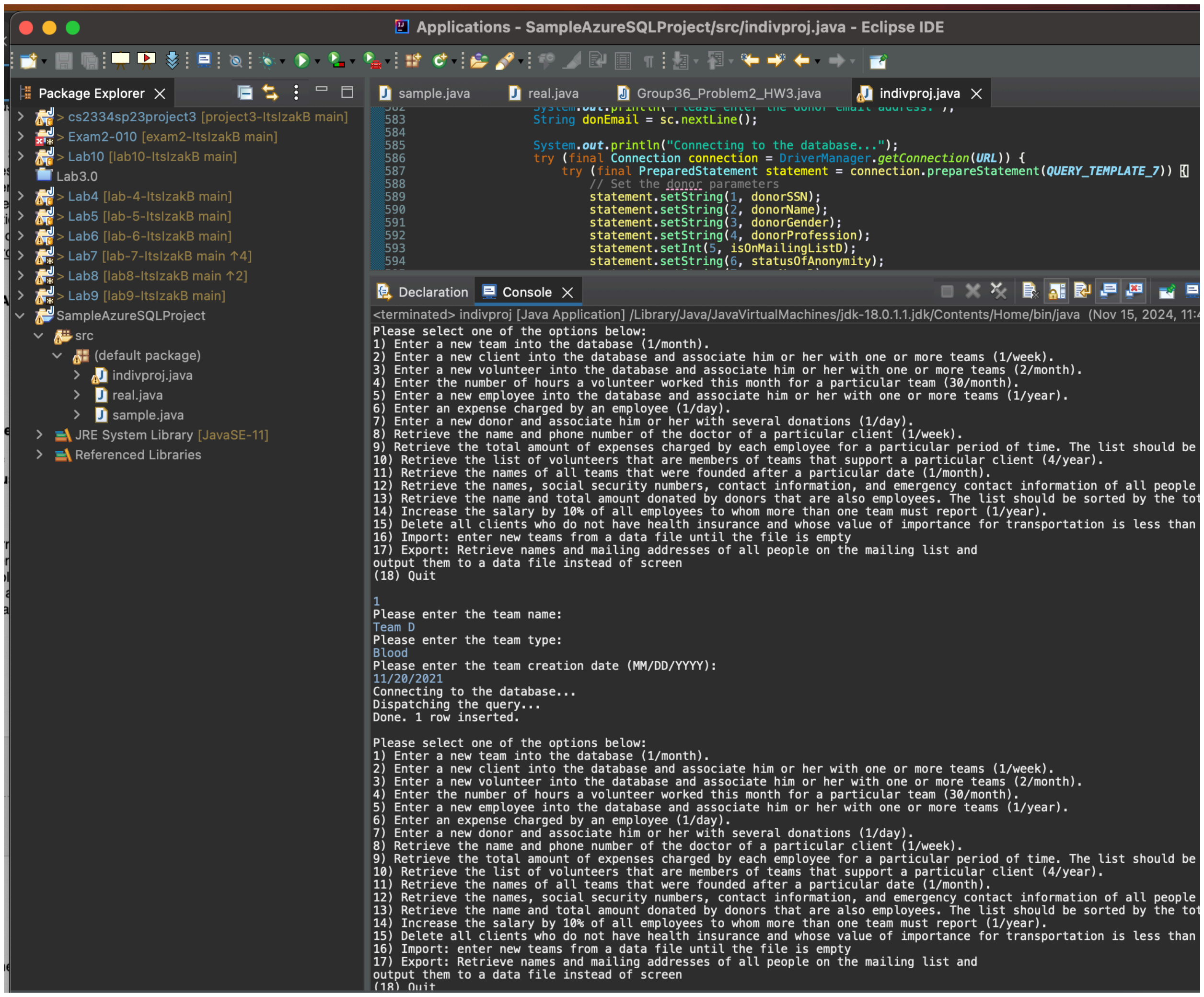
```
    join Volunteer v on p.SSN = v.vol_SSN
    where p.SSN = @client_SSN;
end;

--q11 Retrieve Names of Teams Founded After a Particular Date
go
drop procedure if exists GetTeamsFoundedAfter;
go
create procedure GetTeamsFoundedAfter(
    @date date
)
as
begin
    select team_name
    from Team
    where date > @date;
end;

--q12: Retrieve Names, SSNs, and Emergency Contact Information of All People
go
drop procedure if exists GetAllPeopleWithContactInfo;
go
create procedure GetAllPeopleWithContactInfo
as
begin
    select
        p.name,
        p.SSN,
        ci.mailing_address,
        ci.phone_number,
        ci.email_address,
        ec.name as emergency_contact_name,
        ec.phone_number as emergency_contact_phone,
        ec.relationship
    from Person p
    left join Contact_Information ci on p.SSN = ci.SSN
    left join Emergency_Contact ec on p.SSN = ec.SSN;
end;

--q13: Retrieve Donors Who Are Also Employees
go
drop procedure if exists GetDonorsWhoAreEmployees;
```

```
go
create procedure GetDonorsWhoAreEmployees
as
begin
SELECT p.name,
      (SELECT SUM(amount)
       FROM Donation
       WHERE SSN = d.donr_SSN) AS total_donations
FROM Person p
JOIN Employee e ON p.SSN = e.emp_SSN
JOIN Donor d ON e.emp_SSN = d.donr_SSN;
end;




--q14: Increase Salary for Employees Managing Multiple Teams
go
drop procedure if exists IncreaseSalaryForEmployeesWithMultipleTeams;
go
create procedure IncreaseSalaryForEmployeesWithMultipleTeams
as
begin
   -- ssn of remp
   -- select e.emp_SSN
   -- from Employee e
   -- join Reports_To r on e.emp_SSN = r.employee_SSN
   -- group by e.emp_SSN
   -- having count(distinct r.team_name) > 1;

   -- Update salaries
   update Employee
   set salary = salary * 1.10
   where emp_SSN in(
       select e.emp_SSN
       from Employee e
       join Reports_To r on e.emp_SSN = r.employee_SSN
       group by e.emp_SSN
       having count(distinct r.team_name) > 1
   );
```

```
end;
--q15: Delete clients with no health insurance or trans < 5
GO
DROP PROCEDURE IF EXISTS DeleteClients;
GO

CREATE PROCEDURE DeleteClients
AS
BEGIN
   -- Step 1: Delete dependent rows in Cares_For table
   DELETE FROM Cares_For
   WHERE client_SSN IN
   (
       SELECT c.client_SSN
       FROM Client c
       LEFT JOIN Owns o ON c.client_SSN = o.client_SSN
       LEFT JOIN Insurance_Policy ip ON o.policy_id = ip.policy_id
       WHERE c.transportation_needs < 5 OR ip.insurance_type <> 'health'
   );

   -- Step 2: Delete dependent rows in Owns table
   DELETE FROM Owns
   WHERE client_SSN IN
   (
       SELECT c.client_SSN
       FROM Client c
       LEFT JOIN Owns o ON c.client_SSN = o.client_SSN
       LEFT JOIN Insurance_Policy ip ON o.policy_id = ip.policy_id
       WHERE c.transportation_needs < 5 OR ip.insurance_type <> 'health'
   );

   -- Step 3: Delete from the Client table
   DELETE FROM Client
   WHERE transportation_needs < 5
      OR client_SSN IN (
           SELECT client_SSN
           FROM Owns o
           JOIN Insurance_Policy ip ON o.policy_id = ip.policy_id
           WHERE ip.insurance_type <> 'health'
      );
END;
```

Task 5:

Query1:

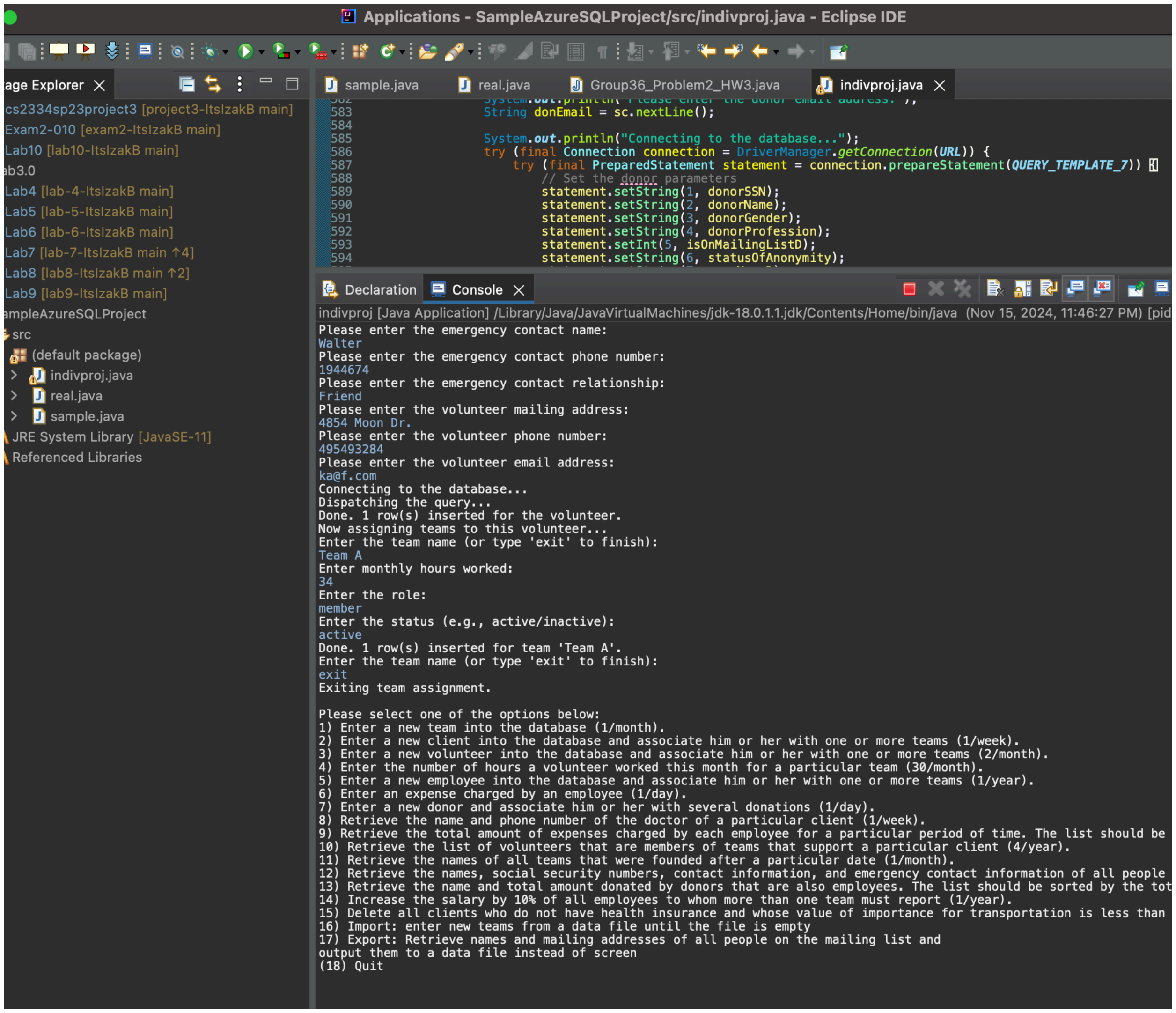

Applications - SampleAzureSQLProject/src/indivproj.java - Eclipse IDE
Package Explorer
> cs2334sp23project3 [project3-ItsIzakB main]
> Exam2-010 [exam2-ItsIzakB main]
> Lab10 [lab10-ItsIzakB main]
Lab3.0
> Lab4 [lab-4-ItsIzakB main]
> Lab5 [lab-5-ItsIzakB main]
> Lab6 [lab-6-ItsIzakB main]
> Lab7 [lab-7-ItsIzakB main ↑4]
> Lab8 [lab8-ItsIzakB main ↑2]
> Lab9 [lab9-ItsIzakB main]
SampleAzureSQLProject
src
(default package)
indivproj.java
real.java
sample.java
JRE System Library [JavaSE-11]
Referenced Libraries
sample.java
real.java
Group36_Problem2_HW3.java
indivproj.java
Declaration
Console

```java
                String donEmail = sc.nextLine();

                System.out.println("Connecting to the database...");
                try (final Connection connection = DriverManager.getConnection(URL)) {
                    try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_7)) {
                        // Set the donor parameters
                        statement.setString(1, donorSSN);
                        statement.setString(2, donorName);
                        statement.setString(3, donorGender);
                        statement.setString(4, donorProfession);
                        statement.setInt(5, isOnMailingListD);
                        statement.setString(6, statusOfAnonymity);
```

```
<terminated> indivproj [Java Application] /Library/Java/JavaVirtualMachines/jdk-18.0.1.1.jdk/Contents/Home/bin/java  (Nov 15, 2024, 11:
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 donor and associate him or her with several donations (1/day).
8) Retrieve the name and phone number of the doctor of a particular client (1/week).
9) Retrieve the total amount of expenses charged by each employee for a particular period of time. The list should be
10) Retrieve the list of volunteers that are members of teams that support a particular client (4/year).
11) Retrieve the names of all teams that were founded after a particular date (1/month).
12) Retrieve the names, social security numbers, contact information, and emergency contact information of all people
13) Retrieve the name and total amount donated by donors that are also employees. The list should be sorted by the tot
14) Increase the salary by 10% of all employees to whom more than one team must report (1/year).
15) Delete all clients who do not have health insurance and whose value of importance for transportation is less than
16) Import: enter new teams from a data file until the file is empty
17) Export: Retrieve names and mailing addresses of all people on the mailing list and
output them to a data file instead of screen
(18) Quit

1
Please enter the team name:
Team D
Please enter the team type:
Blood
Please enter the team creation date (MM/DD/YYYY):
11/20/2021
Connecting to the database...
Dispatching the query...
Done. 1 row inserted.

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 donor and associate him or her with several donations (1/day).
8) Retrieve the name and phone number of the doctor of a particular client (1/week).
9) Retrieve the total amount of expenses charged by each employee for a particular period of time. The list should be
10) Retrieve the list of volunteers that are members of teams that support a particular client (4/year).
11) Retrieve the names of all teams that were founded after a particular date (1/month).
12) Retrieve the names, social security numbers, contact information, and emergency contact information of all people
13) Retrieve the name and total amount donated by donors that are also employees. The list should be sorted by the tot
14) Increase the salary by 10% of all employees to whom more than one team must report (1/year).
15) Delete all clients who do not have health insurance and whose value of importance for transportation is less than
16) Import: enter new teams from a data file until the file is empty
17) Export: Retrieve names and mailing addresses of all people on the mailing list and
output them to a data file instead of screen
(18) Quit
```

Q2:

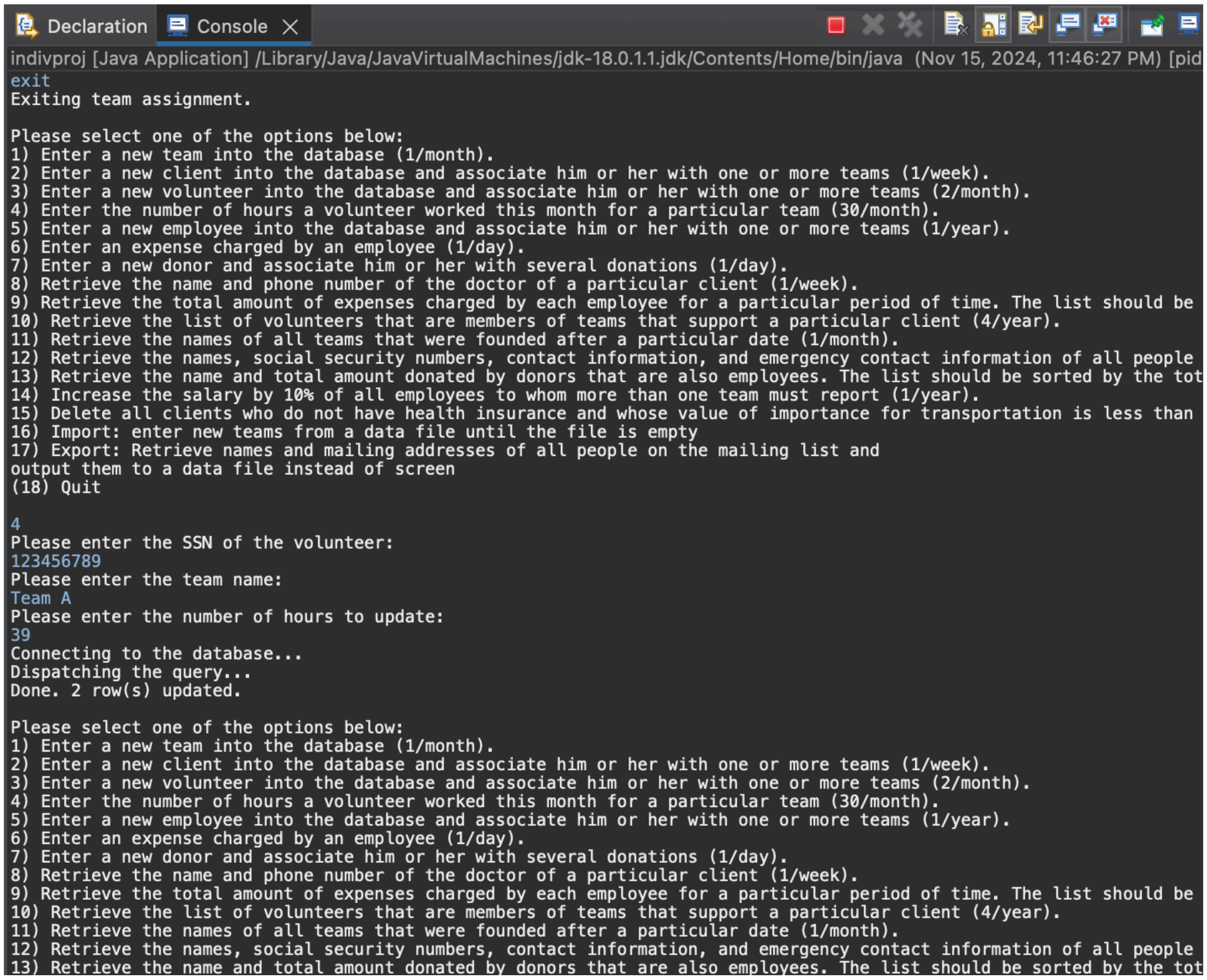
Query 3:



Applications - SampleAzureSQLProject/src/indivproj.java - Eclipse IDE

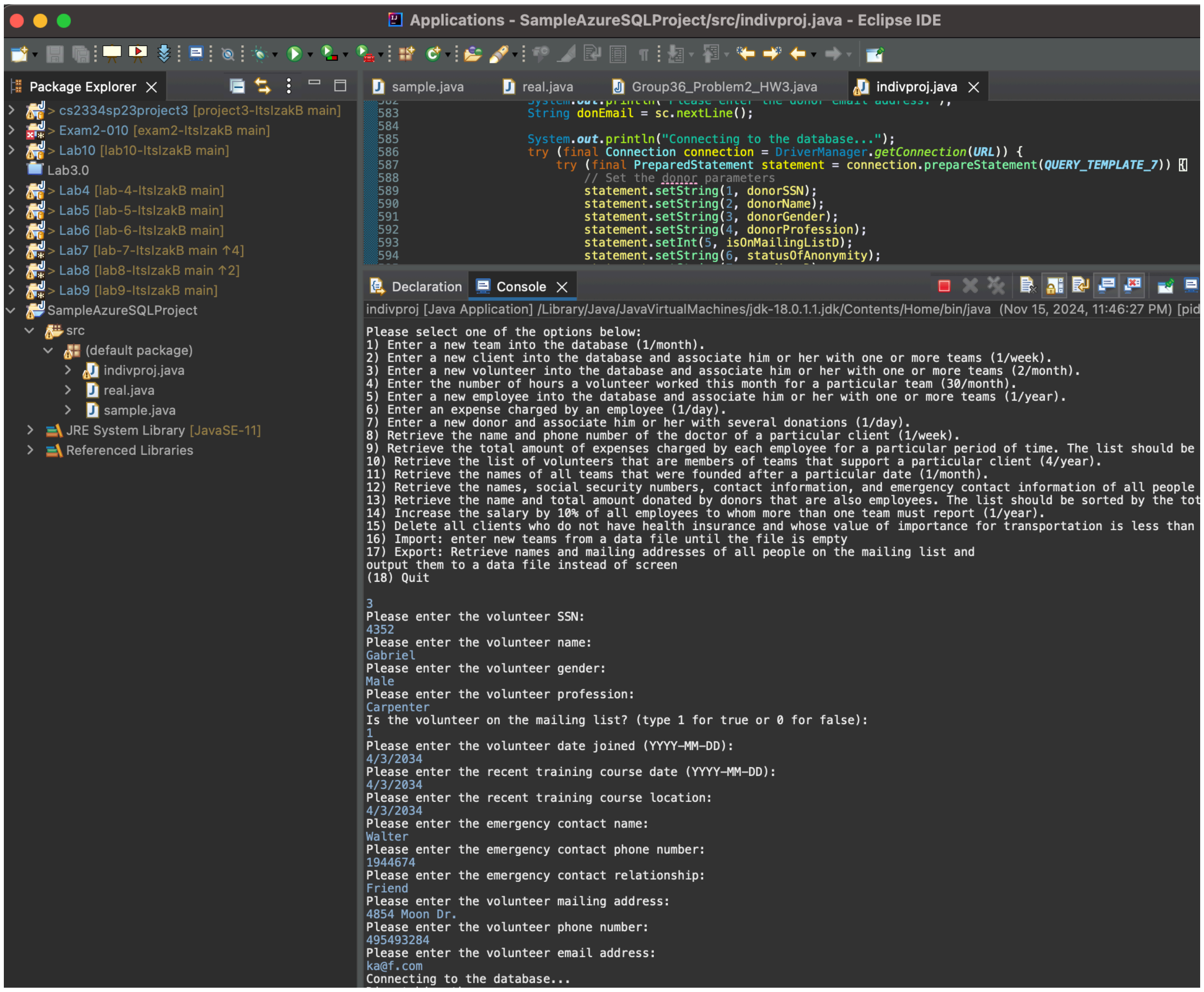
```java
            String donEmail = sc.nextLine();

            System.out.println("Connecting to the database...");
            try (final Connection connection = DriverManager.getConnection(URL)) {
                try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_7)) {
                    // Set the donor parameters
                    statement.setString(1, donorSSN);
                    statement.setString(2, donorName);
                    statement.setString(3, donorGender);
                    statement.setString(4, donorProfession);
                    statement.setInt(5, isOnMailingListD);
                    statement.setString(6, statusOfAnonymity);
```

```
indivproj [Java Application] /Library/Java/JavaVirtualMachines/jdk-18.0.1.1.jdk/Contents/Home/bin/java  (Nov 15, 2024, 11:46:27 PM) [pid
Please enter the emergency contact name:
Walter
Please enter the emergency contact phone number:
1944674
Please enter the emergency contact relationship:
Friend
Please enter the volunteer mailing address:
4854 Moon Dr.
Please enter the volunteer phone number:
495493284
Please enter the volunteer email address:
ka@f.com
Connecting to the database...
Dispatching the query...
Done. 1 row(s) inserted for the volunteer.
Now assigning teams to this volunteer...
Enter the team name (or type 'exit' to finish):
Team A
Enter monthly hours worked:
34
Enter the role:
member
Enter the status (e.g., active/inactive):
active
Done. 1 row(s) inserted for team 'Team A'.
Enter the team name (or type 'exit' to finish):
exit
Exiting team assignment.

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 donor and associate him or her with several donations (1/day).
8) Retrieve the name and phone number of the doctor of a particular client (1/week).
9) Retrieve the total amount of expenses charged by each employee for a particular period of time. The list should be
10) Retrieve the list of volunteers that are members of teams that support a particular client (4/year).
11) Retrieve the names of all teams that were founded after a particular date (1/month).
12) Retrieve the names, social security numbers, contact information, and emergency contact information of all people
13) Retrieve the name and total amount donated by donors that are also employees. The list should be sorted by the tot
14) Increase the salary by 10% of all employees to whom more than one team must report (1/year).
15) Delete all clients who do not have health insurance and whose value of importance for transportation is less than
16) Import: enter new teams from a data file until the file is empty
17) Export: Retrieve names and mailing addresses of all people on the mailing list and
output them to a data file instead of screen
(18) Quit
```



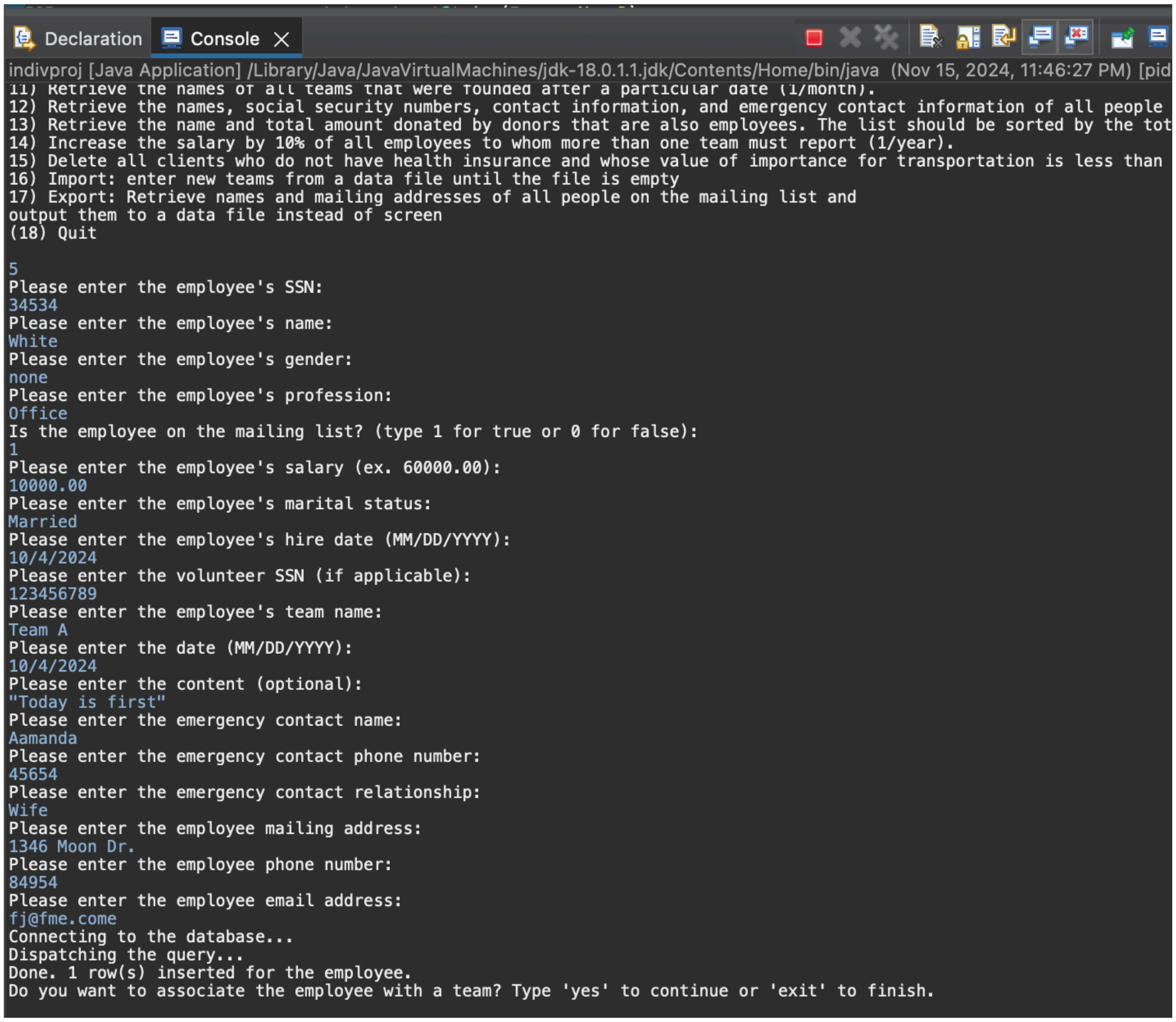
Declaration Console

```
indivproj [Java Application] /Library/Java/JavaVirtualMachines/jdk-18.0.1.1.jdk/Contents/Home/bin/java  (Nov 15, 2024, 11:46:27 PM) [pid
exit
Exiting team assignment.

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 donor and associate him or her with several donations (1/day).
8) Retrieve the name and phone number of the doctor of a particular client (1/week).
9) Retrieve the total amount of expenses charged by each employee for a particular period of time. The list should be
10) Retrieve the list of volunteers that are members of teams that support a particular client (4/year).
11) Retrieve the names of all teams that were founded after a particular date (1/month).
12) Retrieve the names, social security numbers, contact information, and emergency contact information of all people
13) Retrieve the name and total amount donated by donors that are also employees. The list should be sorted by the tot
14) Increase the salary by 10% of all employees to whom more than one team must report (1/year).
15) Delete all clients who do not have health insurance and whose value of importance for transportation is less than
16) Import: enter new teams from a data file until the file is empty
17) Export: Retrieve names and mailing addresses of all people on the mailing list and
output them to a data file instead of screen
(18) Quit

4
Please enter the SSN of the volunteer:
123456789
Please enter the team name:
Team A
Please enter the number of hours to update:
39
Connecting to the database...
Dispatching the query...
Done. 2 row(s) updated.

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 donor and associate him or her with several donations (1/day).
8) Retrieve the name and phone number of the doctor of a particular client (1/week).
9) Retrieve the total amount of expenses charged by each employee for a particular period of time. The list should be
10) Retrieve the list of volunteers that are members of teams that support a particular client (4/year).
11) Retrieve the names of all teams that were founded after a particular date (1/month).
12) Retrieve the names, social security numbers, contact information, and emergency contact information of all people
13) Retrieve the name and total amount donated by donors that are also employees. The list should be sorted by the tot
```

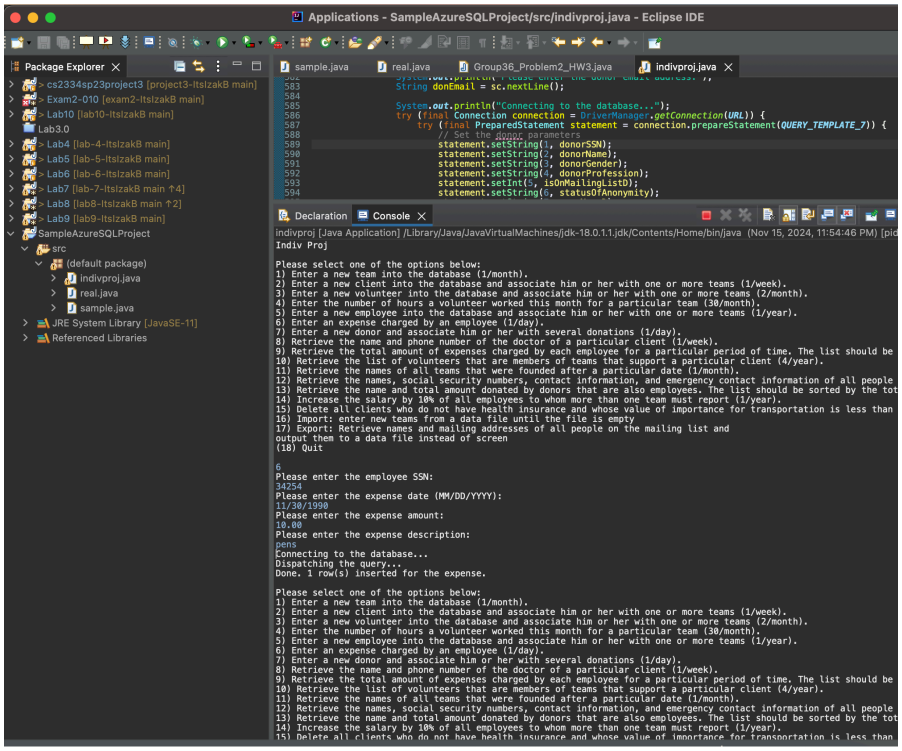

Applications - SampleAzureSQLProject/src/indivproj.java - Eclipse IDE
Package Explorer
> cs2334sp23project3 [project3-ItsIzakB main]
> Exam2-010 [exam2-ItsIzakB main]
> Lab10 [lab10-ItsIzakB main]
Lab3.0
> Lab4 [lab-4-ItsIzakB main]
> Lab5 [lab-5-ItsIzakB main]
> Lab6 [lab-6-ItsIzakB main]
> Lab7 [lab-7-ItsIzakB main ↑4]
> Lab8 [lab8-ItsIzakB main ↑2]
> Lab9 [lab9-ItsIzakB main]
SampleAzureSQLProject
src
(default package)
indivproj.java
real.java
sample.java
JRE System Library [JavaSE-11]
Referenced Libraries
sample.java
real.java
Group36_Problem2_HW3.java
indivproj.java
Declaration
Console

```java
        String donEmail = sc.nextLine();

        System.out.println("Connecting to the database...");
        try (final Connection connection = DriverManager.getConnection(URL)) {
            try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_7)) {
                // Set the donor parameters
                statement.setString(1, donorSSN);
                statement.setString(2, donorName);
                statement.setString(3, donorGender);
                statement.setString(4, donorProfession);
                statement.setInt(5, isOnMailingListD);
                statement.setString(6, statusOfAnonymity);
```

```
indivproj [Java Application] /Library/Java/JavaVirtualMachines/jdk-18.0.1.1.jdk/Contents/Home/bin/java  (Nov 15, 2024, 11:46:27 PM) [pid

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 donor and associate him or her with several donations (1/day).
8) Retrieve the name and phone number of the doctor of a particular client (1/week).
9) Retrieve the total amount of expenses charged by each employee for a particular period of time. The list should be
10) Retrieve the list of volunteers that are members of teams that support a particular client (4/year).
11) Retrieve the names of all teams that were founded after a particular date (1/month).
12) Retrieve the names, social security numbers, contact information, and emergency contact information of all people
13) Retrieve the name and total amount donated by donors that are also employees. The list should be sorted by the tot
14) Increase the salary by 10% of all employees to whom more than one team must report (1/year).
15) Delete all clients who do not have health insurance and whose value of importance for transportation is less than
16) Import: enter new teams from a data file until the file is empty
17) Export: Retrieve names and mailing addresses of all people on the mailing list and
output them to a data file instead of screen
(18) Quit

3
Please enter the volunteer SSN:
4352
Please enter the volunteer name:
Gabriel
Please enter the volunteer gender:
Male
Please enter the volunteer profession:
Carpenter
Is the volunteer on the mailing list? (type 1 for true or 0 for false):
1
Please enter the volunteer date joined (YYYY-MM-DD):
4/3/2034
Please enter the recent training course date (YYYY-MM-DD):
4/3/2034
Please enter the recent training course location:
4/3/2034
Please enter the emergency contact name:
Walter
Please enter the emergency contact phone number:
1944674
Please enter the emergency contact relationship:
Friend
Please enter the volunteer mailing address:
4854 Moon Dr.
Please enter the volunteer phone number:
495493284
Please enter the volunteer email address:
ka@f.com
Connecting to the database...
```

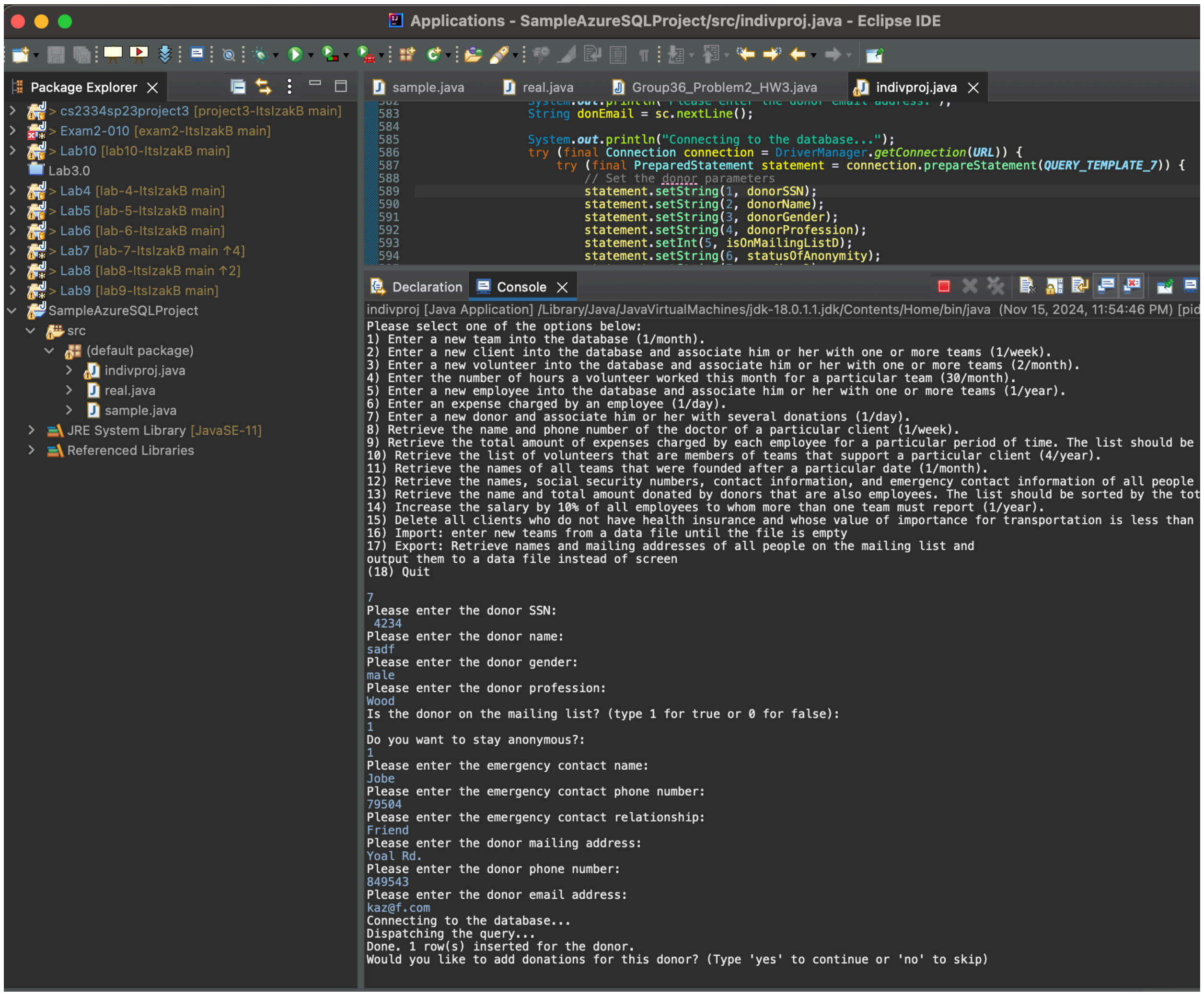


```
indivproj [Java Application] /Library/Java/JavaVirtualMachines/jdk-18.0.1.1.jdk/Contents/Home/bin/java  (Nov 15, 2024, 11:46:27 PM) [pid
11) Retrieve the names of all teams that were founded after a particular date (1/month).
12) Retrieve the names, social security numbers, contact information, and emergency contact information of all people
13) Retrieve the name and total amount donated by donors that are also employees. The list should be sorted by the tot
14) Increase the salary by 10% of all employees to whom more than one team must report (1/year).
15) Delete all clients who do not have health insurance and whose value of importance for transportation is less than
16) Import: enter new teams from a data file until the file is empty
17) Export: Retrieve names and mailing addresses of all people on the mailing list and
output them to a data file instead of screen
(18) Quit

5
Please enter the employee's SSN:
34534
Please enter the employee's name:
White
Please enter the employee's gender:
none
Please enter the employee's profession:
Office
Is the employee on the mailing list? (type 1 for true or 0 for false):
1
Please enter the employee's salary (ex. 60000.00):
10000.00
Please enter the employee's marital status:
Married
Please enter the employee's hire date (MM/DD/YYYY):
10/4/2024
Please enter the volunteer SSN (if applicable):
123456789
Please enter the employee's team name:
Team A
Please enter the date (MM/DD/YYYY):
10/4/2024
Please enter the content (optional):
"Today is first"
Please enter the emergency contact name:
Aamanda
Please enter the emergency contact phone number:
45654
Please enter the emergency contact relationship:
Wife
Please enter the employee mailing address:
1346 Moon Dr.
Please enter the employee phone number:
84954
Please enter the employee email address:
fj@fme.come
Connecting to the database...
Dispatching the query...
Done. 1 row(s) inserted for the employee.
Do you want to associate the employee with a team? Type 'yes' to continue or 'exit' to finish.
```


Applications - SampleAzureSQLProject/src/indivproj.java - Eclipse IDE
Package Explorer
cs2334sp23project3 [project3-ItsIzakB main]
Exam2-010 [exam2-ItsIzakB main]
Lab10 [lab10-ItsIzakB main]
Lab3.0
Lab4 [lab-4-ItsIzakB main]
Lab5 [lab-5-ItsIzakB main]
Lab6 [lab-6-ItsIzakB main]
Lab7 [lab-7-ItsIzakB main ↑4]
Lab8 [lab8-ItsIzakB main ↑2]
Lab9 [lab9-ItsIzakB main]
SampleAzureSQLProject
src
(default package)
indivproj.java
real.java
sample.java
JRE System Library [JavaSE-11]
Referenced Libraries
sample.java
real.java
Group36_Problem2_HW3.java
indivproj.java
Declaration
Console

```java
                String donEmail = sc.nextLine();

                System.out.println("Connecting to the database...");
                try (final Connection connection = DriverManager.getConnection(URL)) {
                    try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_7)) {
                        // Set the donor parameters
                        statement.setString(1, donorSSN);
                        statement.setString(2, donorName);
                        statement.setString(3, donorGender);
                        statement.setString(4, donorProfession);
                        statement.setInt(5, isOnMailingListD);
                        statement.setString(6, statusOfAnonymity);
```

```
indivproj [Java Application] /Library/Java/JavaVirtualMachines/jdk-18.0.1.1.jdk/Contents/Home/bin/java  (Nov 15, 2024, 11:54:46 PM) [pid
Indiv Proj

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 donor and associate him or her with several donations (1/day).
8) Retrieve the name and phone number of the doctor of a particular client (1/week).
9) Retrieve the total amount of expenses charged by each employee for a particular period of time. The list should be
10) Retrieve the list of volunteers that are members of teams that support a particular client (4/year).
11) Retrieve the names of all teams that were founded after a particular date (1/month).
12) Retrieve the names, social security numbers, contact information, and emergency contact information of all people
13) Retrieve the name and total amount donated by donors that are also employees. The list should be sorted by the tot
14) Increase the salary by 10% of all employees to whom more than one team must report (1/year).
15) Delete all clients who do not have health insurance and whose value of importance for transportation is less than
16) Import: enter new teams from a data file until the file is empty
17) Export: Retrieve names and mailing addresses of all people on the mailing list and
output them to a data file instead of screen
(18) Quit

6
Please enter the employee SSN:
34254
Please enter the expense date (MM/DD/YYYY):
11/30/1990
Please enter the expense amount:
10.00
Please enter the expense description:
pens
Connecting to the database...
Dispatching the query...
Done. 1 row(s) inserted for the expense.

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 donor and associate him or her with several donations (1/day).
8) Retrieve the name and phone number of the doctor of a particular client (1/week).
9) Retrieve the total amount of expenses charged by each employee for a particular period of time. The list should be
10) Retrieve the list of volunteers that are members of teams that support a particular client (4/year).
11) Retrieve the names of all teams that were founded after a particular date (1/month).
12) Retrieve the names, social security numbers, contact information, and emergency contact information of all people
13) Retrieve the name and total amount donated by donors that are also employees. The list should be sorted by the tot
14) Increase the salary by 10% of all employees to whom more than one team must report (1/year).
15) Delete all clients who do not have health insurance and whose value of importance for transportation is less than
```

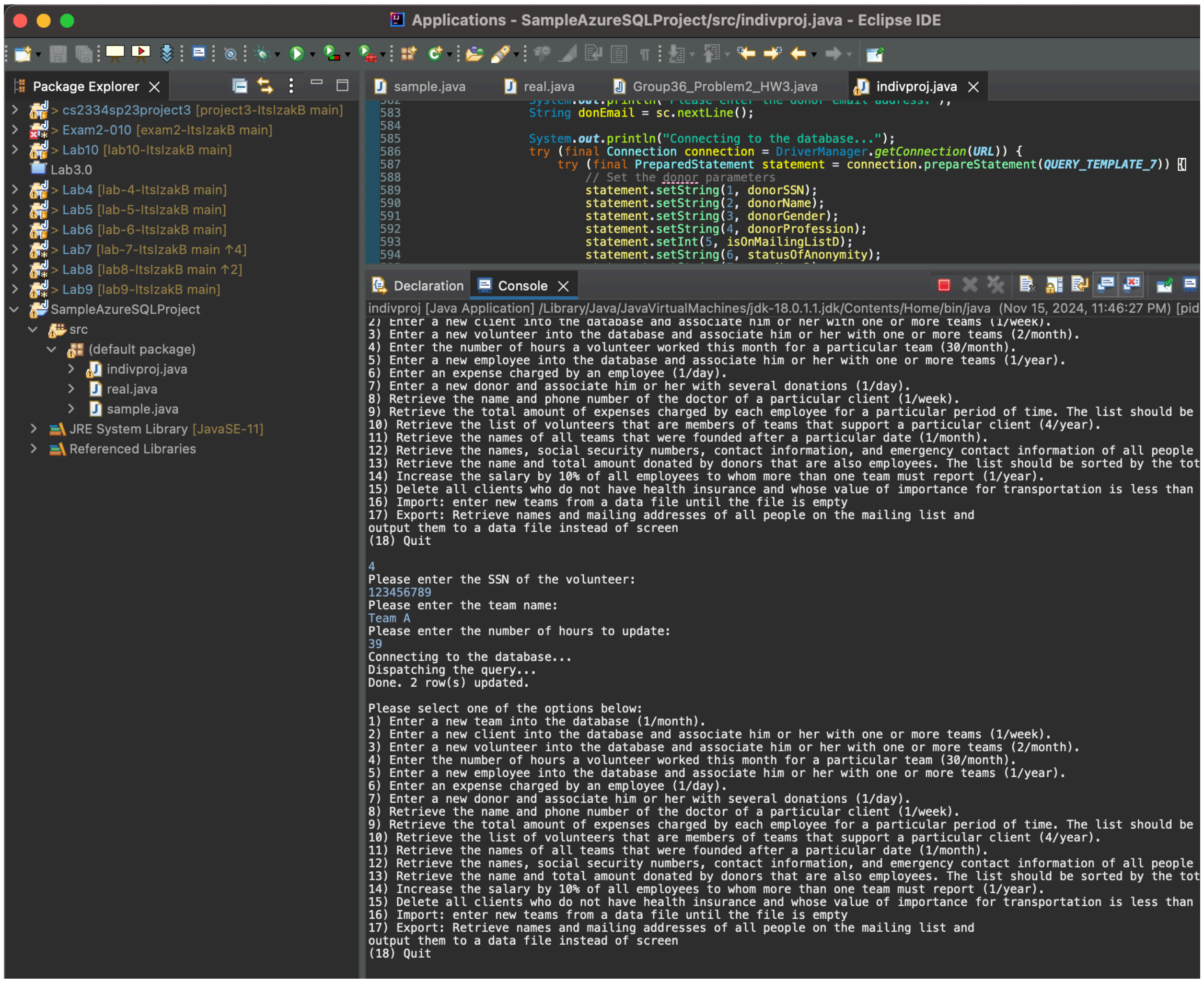


Applications - SampleAzureSQLProject/src/indivproj.java - Eclipse IDE
Package Explorer
> cs2334sp23project3 [project3-ItsIzakB main]
> Exam2-010 [exam2-ItsIzakB main]
> Lab10 [lab10-ItsIzakB main]
Lab3.0
> Lab4 [lab-4-ItsIzakB main]
> Lab5 [lab-5-ItsIzakB main]
> Lab6 [lab-6-ItsIzakB main]
> Lab7 [lab-7-ItsIzakB main ↑4]
> Lab8 [lab8-ItsIzakB main ↑2]
> Lab9 [lab9-ItsIzakB main]
SampleAzureSQLProject
src
(default package)
indivproj.java
real.java
sample.java
JRE System Library [JavaSE-11]
Referenced Libraries
sample.java
real.java
Group36_Problem2_HW3.java
indivproj.java
Declaration
Console

```
        String donEmail = sc.nextLine();

        System.out.println("Connecting to the database...");
        try (final Connection connection = DriverManager.getConnection(URL)) {
            try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_7)) {
                // Set the donor parameters
                statement.setString(1, donorSSN);
                statement.setString(2, donorName);
                statement.setString(3, donorGender);
                statement.setString(4, donorProfession);
                statement.setInt(5, isOnMailingListD);
                statement.setString(6, statusOfAnonymity);
```

indivproj [Java Application] /Library/Java/JavaVirtualMachines/jdk-18.0.1.1.jdk/Contents/Home/bin/java (Nov 15, 2024, 11:54:46 PM) [pid

```
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 donor and associate him or her with several donations (1/day).
8) Retrieve the name and phone number of the doctor of a particular client (1/week).
9) Retrieve the total amount of expenses charged by each employee for a particular period of time. The list should be
10) Retrieve the list of volunteers that are members of teams that support a particular client (4/year).
11) Retrieve the names of all teams that were founded after a particular date (1/month).
12) Retrieve the names, social security numbers, contact information, and emergency contact information of all people
13) Retrieve the name and total amount donated by donors that are also employees. The list should be sorted by the tot
14) Increase the salary by 10% of all employees to whom more than one team must report (1/year).
15) Delete all clients who do not have health insurance and whose value of importance for transportation is less than
16) Import: enter new teams from a data file until the file is empty
17) Export: Retrieve names and mailing addresses of all people on the mailing list and
output them to a data file instead of screen
(18) Quit

7
Please enter the donor SSN:
 4234
Please enter the donor name:
sadf
Please enter the donor gender:
male
Please enter the donor profession:
Wood
Is the donor on the mailing list? (type 1 for true or 0 for false):
1
Do you want to stay anonymous?:
1
Please enter the emergency contact name:
Jobe
Please enter the emergency contact phone number:
79504
Please enter the emergency contact relationship:
Friend
Please enter the donor mailing address:
Yoal Rd.
Please enter the donor phone number:
849543
Please enter the donor email address:
kaz@f.com
Connecting to the database...
Dispatching the query...
Done. 1 row(s) inserted for the donor.
Would you like to add donations for this donor? (Type 'yes' to continue or 'no' to skip)
```


Applications - SampleAzureSQLProject/src/indivproj.java - Eclipse IDE
Package Explorer
> cs2334sp23project3 [project3-ItsIzakB main]
> Exam2-010 [exam2-ItsIzakB main]
> Lab10 [lab10-ItsIzakB main]
Lab3.0
> Lab4 [lab-4-ItsIzakB main]
> Lab5 [lab-5-ItsIzakB main]
> Lab6 [lab-6-ItsIzakB main]
> Lab7 [lab-7-ItsIzakB main ↑4]
> Lab8 [lab8-ItsIzakB main ↑2]
> Lab9 [lab9-ItsIzakB main]
SampleAzureSQLProject
src
(default package)
indivproj.java
real.java
sample.java
JRE System Library [JavaSE-11]
Referenced Libraries
sample.java
real.java
Group36_Problem2_HW3.java
indivproj.java
Declaration
Console

```java
            String donEmail = sc.nextLine();

            System.out.println("Connecting to the database...");
            try (final Connection connection = DriverManager.getConnection(URL)) {
                try (final PreparedStatement statement = connection.prepareStatement(QUERY_TEMPLATE_7)) {
                    // Set the donor parameters
                    statement.setString(1, donorSSN);
                    statement.setString(2, donorName);
                    statement.setString(3, donorGender);
                    statement.setString(4, donorProfession);
                    statement.setInt(5, isOnMailingListD);
                    statement.setString(6, statusOfAnonymity);
```

indivproj [Java Application] /Library/Java/JavaVirtualMachines/jdk-18.0.1.1.jdk/Contents/Home/bin/java (Nov 15, 2024, 11:46:27 PM) [pid

```
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 donor and associate him or her with several donations (1/day).
8) Retrieve the name and phone number of the doctor of a particular client (1/week).
9) Retrieve the total amount of expenses charged by each employee for a particular period of time. The list should be
10) Retrieve the list of volunteers that are members of teams that support a particular client (4/year).
11) Retrieve the names of all teams that were founded after a particular date (1/month).
12) Retrieve the names, social security numbers, contact information, and emergency contact information of all people
13) Retrieve the name and total amount donated by donors that are also employees. The list should be sorted by the tot
14) Increase the salary by 10% of all employees to whom more than one team must report (1/year).
15) Delete all clients who do not have health insurance and whose value of importance for transportation is less than
16) Import: enter new teams from a data file until the file is empty
17) Export: Retrieve names and mailing addresses of all people on the mailing list and
output them to a data file instead of screen
(18) Quit

4
Please enter the SSN of the volunteer:
123456789
Please enter the team name:
Team A
Please enter the number of hours to update:
39
Connecting to the database...
Dispatching the query...
Done. 2 row(s) updated.

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 donor and associate him or her with several donations (1/day).
8) Retrieve the name and phone number of the doctor of a particular client (1/week).
9) Retrieve the total amount of expenses charged by each employee for a particular period of time. The list should be
10) Retrieve the list of volunteers that are members of teams that support a particular client (4/year).
11) Retrieve the names of all teams that were founded after a particular date (1/month).
12) Retrieve the names, social security numbers, contact information, and emergency contact information of all people
13) Retrieve the name and total amount donated by donors that are also employees. The list should be sorted by the tot
14) Increase the salary by 10% of all employees to whom more than one team must report (1/year).
15) Delete all clients who do not have health insurance and whose value of importance for transportation is less than
16) Import: enter new teams from a data file until the file is empty
17) Export: Retrieve names and mailing addresses of all people on the mailing list and
output them to a data file instead of screen
(18) Quit
```