

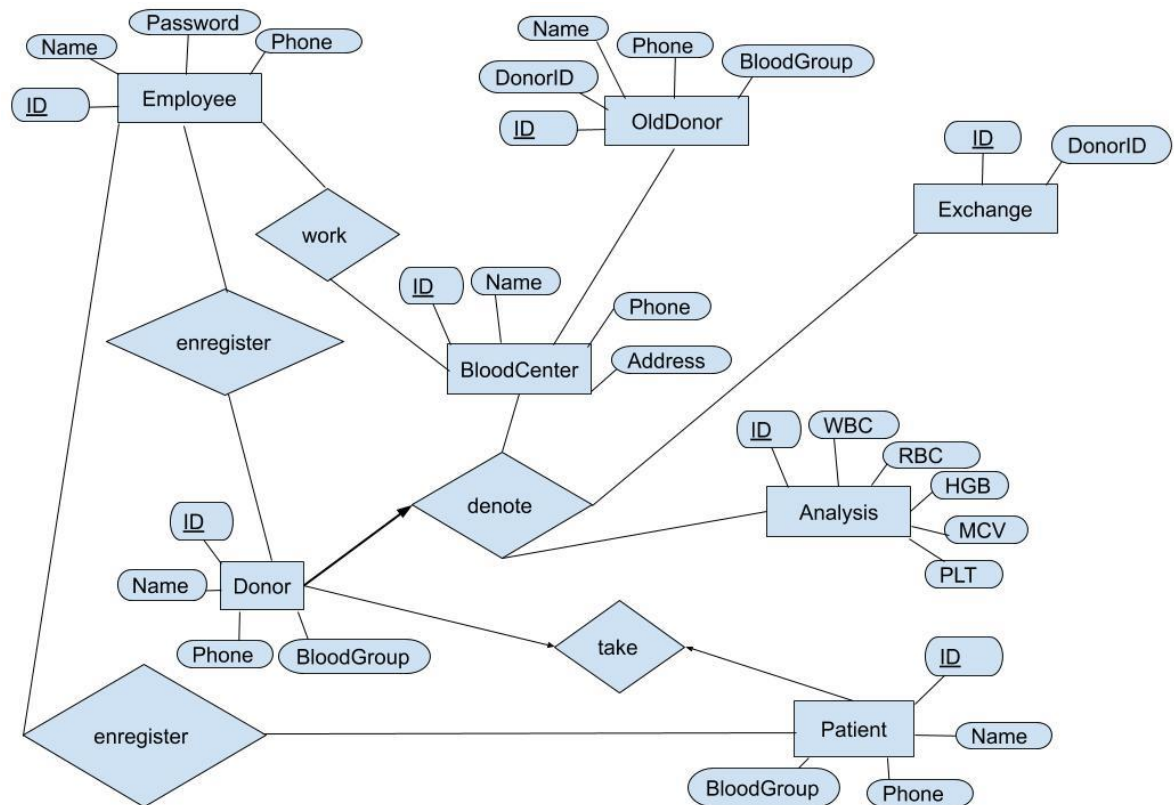
CS 350 – DATABASE SYSTEMS
TERM PROJECT
PART 3: Design and Implementation
Muhammet Mesut Koç – Blood Bank

1. Motivation and Requirements:

- This application allows people who need blood to easily find blood in Istanbul.
- The system has a lot of donors, when the application is run, the person who needs the blood is registered and at the same time the needed blood can be searched. However, the analysis results of the blood taken can be viewed from the system.

2. Conceptual Database Design

- In my project, I need patient who need blood and donor who donate blood and blood center to exchange blood. For analysis of blood I use random method to create values.
-



3. Logical Database Design

- **Tables**

```
CREATE TABLE BloodCenter ( CenterID FLOAT,  
                             CenterName NCHAR(150),
```

CenterPhone NCHAR(20),
CenterAddress NCHAR(150),
PRIMARY KEY (CenterID))

CREATE TABLE Donor (DonorID FLOAT,
DonorName NVARCHAR(255),
DonorPhone FLOAT,
DonorBloodGroup NVARCHAR(255),
CenterID FLOAT,
PRIMARY KEY (DonorID),
FOREIGN KEY (CenterID) REFERENCES BloodCenter)

CREATE TABLE Patient (PatientID FLOAT,
PatientName NCHAR(30),
PatientPhone FLOAT,
PatientBloodGroup NCHAR(5),
PRIMARY KEY (PatientID))

CREATE TABLE Employee (EmployeeID FLOAT,
EmployeeName NCHAR(30),
EmployeePassword NCHAR(20),
EmployeePhone FLOAT,
CenterID FLOAT,
PRIMARY KEY (EmployeeID),
FOREIGN KEY (CenterID) REFERENCES BloodCenter)

CREATE TABLE OldDonor (ID NUMERIC (18,0),
DonorID FLOAT,
DonorName NCHAR(30),
DonorPhone FLOAT,
DonorBloodGroup NCHAR(5),
CenterID FLOAT,
PRIMARY KEY (ID),
FOREIGN KEY (CenterID) REFERENCES BloodCenter)

CREATE TABLE Exchange (ExchangeID NUMERIC(18,0),
PatientID FLOAT,
DonorID FLOAT,
CenterID FLOAT,
EmployeeID FLOAT,
PRIMARY KEY (ExchangeID),
FOREIGN KEY (PatientID) REFERENCES Patient,
FOREIGN KEY (CenterID) REFERENCES BloodCenter,
FOREIGN KEY (EmployeeID) REFERENCES Employee)

```

CREATE TABLE Analysis ( AnalysisID NUMERIC(18,0),
                        ExchangeID NUMERIC(18,0),
                        WBC NCHAR (10),
                        RBC NCHAR (10),
                        HGB NCHAR (10),
                        MCV NCHAR (10),
                        PLT NCHAR (10),
                        PRIMARY KEY (AnalysisID),
                        FOREIGN KEY (ExchangeID) REFERENCES Exchange)

```

- **Functional Dependencies**

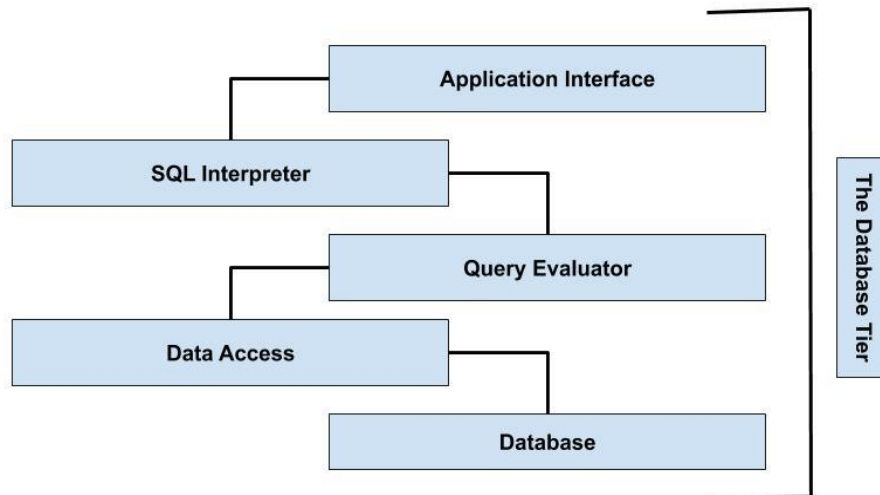
1. CenterID \longrightarrow CenterName, CenterAddress, CenterPhone
2. CenterName, CenterAddress \longrightarrow CenterID
3. PatientID \longrightarrow PatientName, PatientPhone, PatientBloodGroup
4. PatientName, PatientPhone \longrightarrow PatientID
5. DonorID \longrightarrow DonorName, DonorPhone, DonorBloodGroup, CenterID
6. DonorName, DonorPhone \longrightarrow DonorBloodGroup
7. DonorName, DonorPhone, DonorBloodGroup \longrightarrow DonorID
8. EmployeeID \longrightarrow EmployeeName, EmployeePhone, CenterID
9. EmployeeName, EmployeePhone \longrightarrow EmployeeID
10. ExchangeID \longrightarrow PatientID, DonorID, CenterID
11. AnalysisID \longrightarrow ExchangeID, WBC, RBC, HGB, MVC, PLT

- **Normalization**

- 1- It is in BCNF because it has primary and candidate key.
- 3- It is in BCNF because of having primary and candidate key.
- 5- There are primary and candidate key that refers CenterID, so it is in BCNF.
- 8- It is in BCNF because it has candidate key and primary key and doesn't have trivial dependency.
- 10- It is in BCNF because it has primary key and candidate key that refers DonorID, PatientID and CenterID, also there is no trivial dependency.
- 11- It is in BCNF because it has primary key and candidate key.

4. Application Design and Implementation

- Tiers and Connections



Application Interface: Libraries for communicating with the DBMS.

SQL Interpreter: A parser that checks the syntax of incoming query statements.

Query Evaluator: Generates different plans for evaluating a query by considering database statistics and properties, selects one of these plans, and translates the plan into low-level actions that are executed.

Data Access: The modules that manage access to the data stored on disk.

Database: The physical data itself stored in data files.

- Queries in Project

Query: **Select CenterName
from BloodCenter**

Semantics: List all center names that are we use.

Query: **Insert into
Donor(DonorID,DonorName,DonorPhone,DonorBloodGroup,
CenterID) Values(@ID,@Name,@Phone,@BloodGroup,@ CenterID)**

Semantics: Insert datas that donor information with value which are named.

Query: **Select CenterID
from BloodCenter where CenterName= '' +
DonorCentercomboBox1.SelectedItem.ToString() + ''''**

Semantics: List center id's of blood center that have same center name with chosen by user input in listbox.

Query: **Insert into**
Employee (EmployeeID, EmployeeName, EmployeePassword,
EmployeePhone, CenterID)
Values(@ID, @Name, @Password, @Phone, @CenterID)

Semantics: Insert datas that employee information with value which are named.

Query: **Select ***
from Donor
where DonorID = '' + deleteidtextBox1.Text + '' and DonorName = '' +
deletenametextBox2.Text + '' and DonorPhone = '' +
deletephonetextBox3.Text + ''

Semantics: List all donors that denote with same id, name and phone with user input.

Query: **Delete**
from Donor
where DonorName = ''
+ listBox1.SelectedItem.ToString() + ''

Semantics: Delete donor which is choosen by user from listbox.

Query: **Select EmployeeName, EmployeePassword**
from Employee
where EmployeeName = '' + UsertextBox.Text + '' and
EmployeePassword = '' + PasswordtextBox.Text + ''

Semantics: List employee name and password that are equal with user input from textbox.

Query: **Select ***
from Donor
where DonorBloodGroup = '' + BloodtextBox.Text + ''
Semantics: List all donor that has same blood type with user input.

Query: **Select ***
from Donor
where DonorName = '' + donorlistBox1.SelectedItem.ToString() + ''

Semantics: Select all donor that has same name with user selected item in listbox.

Query: **Select CenterName**
from BloodCenter where CenterID = '' + dr["CenterID"] + ''

Semantics: List center names that are had same center id with user input.

Query: **Insert into**
Patient (PatientID, PatientName, PatientPhone, PatientBloodGroup)
Values (@ID, @Name, @Phone, @Blood)

Semantics: Insert datas that patient information with value which are named.

Query: **Insert into**
Exchange (PatientID, DonorID,EmployeeID, CenterID)
Values (@Pati, @Donor, @Employee, @Center)

Semantics: Insert datas that exchange information with value which are named.

Query: **Select DonorId**
from Donor
where DonorName= ''
+ donorlistBox1.SelectedItem.ToString() + ''

Semantics: List donor id of donors that are selected from listbox with user input.

Query: **Select EmployeeID,CenterID**
from Employee
where EmployeeName= '' + Form1.user+ ''

Semantics: List employee id and center id that are same value selected itemfrom listbox with employee name.

Query: **Insert into**
OldDonor(DonorID,DonorName,DonorPhone,DonorBloodGroup,
CenterID)
Values (@ID,@Name,@Phone,@Blood,@ Center)

Semantics: Insert datas that old donor which are denote, information with value which are named.

Query: **Select**
DonorID,DonorName,DonorPhone,DonorBloodGroup,CenterID
from Donor
where DonorName ='' + donorlistBox1.SelectedItem.ToString() + ''

Semantics: List donor id, name, phone, blood type and center id that have same value selected item from listbox with donor name.

Query: **Delete**
from Donor
where DonorName=''
+ donorlistBox1.SelectedItem.ToString() + ''

Semantics: Delete donor which that has same name with choosen by user from listbox.

Query: **Insert into
Analysis(ExchangeID,WBC,RBC,HGB,MCV,PLT)
Values(@ex,@wbc,@rbc,@hgb,@mcv,@plt)**

Semantics: : Insert datas that analysis result which are information with value which are named.

Query: **Select ExchangeID
from Exchange
where PatientID='' + IDtextBox1.Text+''**

Semantics: List exchange id that has same patient id with user input.

Query: **Select *
from OldDonor
where DonorID = '' + OldDonorIDtextBox1.Text + ''**

Semantics: List all old donors that have same donor id with user input.

Query: **Select DonorID
from OldDonor
where DonorName='' + OldDonorlistBox1.SelectedItem.ToString() + ''**

Semantics: List donor id's that have same donor name with user input from listbox.

Query: **Select Exchange.ExchangeID
from Exchange
Inner Join OldDonor On
Exchange.DonorID='' + dr["DonorID"].ToString()+''**

Semantics: List exchange id's that have same donor id with exchange and old donor.

Query: **Select Analysis.WBC, Analysis.RBC, Analysis.HGB,
Analysis.MCV, Analysis.PLT
from Analysis
Inner Join Exchange On Analysis.ExchangeID='' +
dr2["ExchangeID"].ToString() + ''**

Semantics: List analysis result that has same exchange id with exchange id and analysis id.

- **Program Language**

I use C# programming language and develop windows form application. Also, I use Microsoft SQL Server Management Studio to create database.

- **Data Source**

I fetch data from web page (<https://www.istanbul.net.tr/kent-rehberi/saglik-kuruluslari/kan-merkezleri/5/170/1>) that contains blood center information which are name, phone and address. Secondly, I use excel file that I got it from kaggle.com and I change and add some information on this file.

- **Features of Application**

The image displays four overlapping windows from a Windows Forms application:

- Form1 (Login):** Features a red circular logo with a white person icon. It has input fields for 'Username' (containing 'mesut koc') and 'Password' (containing '123456'), and a 'Login' button.
- MainWindow (Find Blood for Patient):** Titled 'Find Blood for Patient' in red. It has tabs for 'Patient' and 'Donor'. The 'Patient' tab has fields for ID, Name, Phone Number, and Blood Group, with a 'Find Donor' button. The 'Donor' tab has a large empty box and buttons for 'Select Donor' and 'Analysis'. To the right, labels for 'Donor ID', 'Phone Number', 'Blood Type', and 'Center Name' are visible.
- CreateAccount (Employee Registration):** Titled 'Employee Registration'. It has fields for ID, Username, Password, Phone, and a dropdown for Center, with an 'Add Info' button.
- AddDonor (Donor Registration):** Titled 'Donor Registration'. It has fields for Donor ID, Donor Name, Phone Number, Blood Group, and a dropdown for Center Name, with an 'Add Info' button.

In this picture, contains forms that employee and donor registration. Also, main window of my application that is find blood for patient.

The image shows two overlapping windows from a desktop application. The top window, titled 'DeleteDonor', has a light gray background and contains the title 'Cancel Donor' in bold black text. Below the title are three input fields: 'ID' (a single box), 'Name' (two adjacent boxes), and 'Phone' (a single box). At the bottom are two buttons labeled 'Find' and 'Delete'. The bottom window, titled 'OldDonorResult', has a white background and contains the title 'Find Analysis Result' in bold red text. It features a 'Donor ID' label above a single input box, followed by a 'Find' button. To the right of the input box is a large empty rectangular area. Further right is a table with two columns: the first column lists blood components (WBC, RBC, HGB, MCV, PLT) and the second column lists the word 'Result' for each component.

Find Analysis Result	
Donor ID	
Find	
WBC	Result
RBC	Result
HGB	Result
MCV	Result
PLT	Result

This picture contains two forms of my application. These are cancel to donor and showing blood analysis result.