BLOOD DONATION DATABASE PROJECT

Nour Kahile

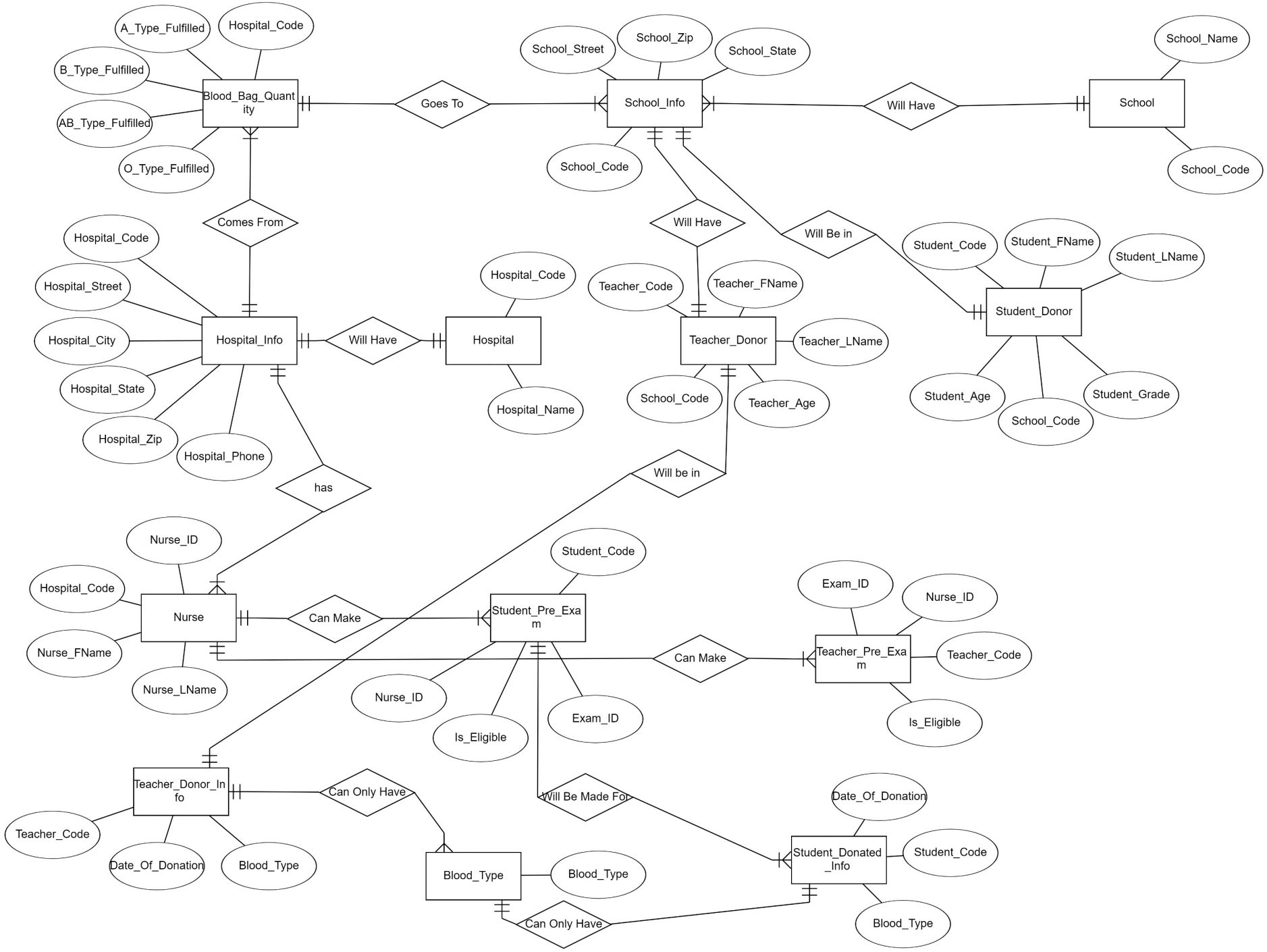
Database systems

Database Name: FINAL52

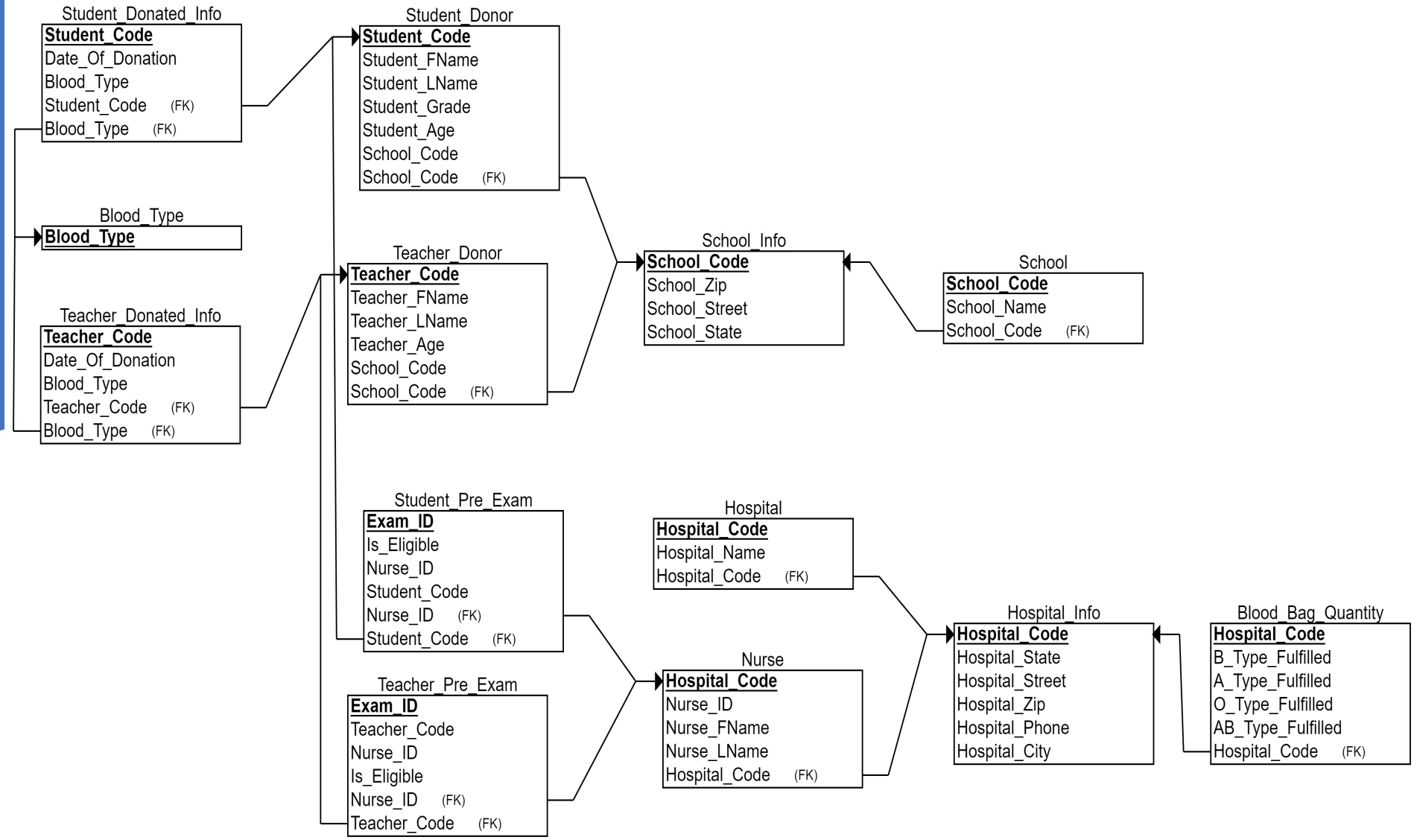
Enterprise Statement:

For my final project, I worked on implementing a database to help and assist in the blood donation process in schools, to go further with the project and to put the database to use I implemented a program in java that will interact with the database.

Fixed and Updated ERD



Relational Schema &Dependency Diagrams



Database Implemented in 3NF

-The whole database is implemented **in the third normal form (3NF)** by reducing the complexity of tables and reducing the redundancy of data, For example let's look at the Student_Pre_Exam table and Teacher_Pre_Exam table, Both could be combined In on table but that will result in repeating data and a high chance in repeating ID's

Function Of The Most Relevant Entities

*The tables with **Donated Individuals** and **School Students** will have an auto increment so whenever we insert new data it will increment the previous inserted row ID by one.

- **Student_Donor:** Will hold student information.
- **Teacher_Donor**: Will hold teacher information.
- **School_Info:** Will hold all school information.
- **School:** School unique code and the name of the school.
- **Hospital_Info:** Will hold hospital information.
- **Nurse:** Will have all the Nurses information.

Function Of The Most Relevant Entities

- **Student_Pre_Exam:** Will have the student code, nurse code, and whether the student is eligible to donate.
- **Teacher_Pre_Exam:** Will have the student code, nurse code, and whether the teacher is eligible to donate.
- **Blood_Bag_Quantity:** This table will have the quantity needed for each blood type
- **Blood Type:** Will have all the blood types (A,B,AB,O)
- **Student_Donated_Info:** Will have all the information about the students that have successfully donated
- **Teacher_Donated_Info:** Will have all the information about the teachers that have successfully donated

Operations On The Database

- **Student_Update_Quantity Trigger**: This trigger will execute after insert on the **Student_Donated_Info** Table, it will check what blood type has been inserted, then it will update the **Blood_Bag_Quantity**, and decrease the required blood type quantity by one, here's a screenshot for better explanation

```
    CREATE TRIGGER Student_Update_Quantity
    ON Student_Donated_Info
    AFTER INSERT AS
    BEGIN
    DECLARE @Type CHAR
    SELECT TOP 1 @Type = Blood_Type FROM Student_Donated_Info ORDER BY Student_Code DESC
        IF (@Type) = 'A'

            UPDATE Blood_Bag_Quantity
                SET A_Type_Fulfilled = A_Type_Fulfilled-1;

        IF (@Type) = 'B'

            UPDATE Blood_Bag_Quantity
                SET B_Type_Fulfilled = B_Type_Fulfilled-1;

        IF (@Type) = 'AB'

            UPDATE Blood_Bag_Quantity
                SET AB_Type_Fulfilled = AB_Type_Fulfilled-1;

        IF (@Type) = 'O'

            UPDATE Blood_Bag_Quantity
                SET O_Type_Fulfilled = O_Type_Fulfilled-1;

    END
GO

----------------------------------------
Drop TRIGGER Update_Quantity;
----------------------------------------
```

```
267    INSERT INTO Student_Donated_Info VALUES(2, '02-06-2021','O')
268
```

Messages

```
9:01:57 PM     Started executing query at Line 268
               (1 row affected)
               (1 row affected)
               Total execution time: 00:00:00.050
```

```
146    --------------------
147    CREATE TABLE Blood_Bag_Quantity(
148        Hospital_Code INT NOT NULL,
149        A_Type_Fulfilled INT,
150        B_Type_Fulfilled INT,
151        AB_Type_Fulfilled INT,
152        O_Type_Fulfilled INT,
153        PRIMARY KEY(Hospital_Code),
154        FOREIGN KEY(Hospital_Code) REFERENCES Hospital_Info(Hospital_Code)
155    );
156    --------------------
157    INSERT INTO Blood_Bag_Quantity VALUES(1,50,50,50,50);
158    --------------------
159    --------------------
160    SELECT * FROM Blood_Bag_Quantity;
161    drop TABLE Blood_Bag_Quantity;
162    --------------------
```

Results Messages

	Hospital_Code	A_Type_Fulfilled	B_Type_Fulfilled	AB_Type_Fulfilled	O_Type_Fulfilled
1	1	34	48	50	40

Operations On The Database

-**Teacher_Update_Quantity:** This trigger similar to the previous on but for Teachers, It will execute after insert on **Teacher_Donated_Info** Table, it will check what blood type has been inserted, then it will update the Blood_Bag_Quantity, and decrease the required blood type quantity by one, here's a screenshot for better exlaination

```
CREATE TRIGGER Teacher_Update_Quantity
ON Teacher_Donated_Info
AFTER INSERT AS
BEGIN
DECLARE @Type CHAR
SELECT TOP 1 @Type = Blood_Type FROM Teacher_Donated_Info ORDER BY Teacher_Code DESC
    IF (@Type) = 'A'

        UPDATE Blood_Bag_Quantity
            SET A_Type_Fulfilled = A_Type_Fulfilled-1;

    IF (@Type) = 'B'

        UPDATE Blood_Bag_Quantity
            SET B_Type_Fulfilled = B_Type_Fulfilled-1;

    IF (@Type) = 'AB'

        UPDATE Blood_Bag_Quantity
            SET AB_Type_Fulfilled = AB_Type_Fulfilled-1;

    IF (@Type) = 'O'

        UPDATE Blood_Bag_Quantity
            SET O_Type_Fulfilled = O_Type_Fulfilled-1;

END
GO
-------------------------------------
DROP TRIGGER Teacher_Update_Quantity;
-------------------------------------
```

```sql
CREATE TABLE Teacher_Donated_Info(
    Teacher_Code INT NOT NULL,
    Date_Of_Donation DATE NOT NULL,
    Blood_Type char(3),
    PRIMARY KEY(Teacher_Code),
    FOREIGN KEY(Teacher_Code) REFERENCES Teacher_Donor(Teacher_Code),
    FOREIGN KEY(Blood_Type) REFERENCES Blood_Type(Blood_Type)
);
--------------------
SELECT * FROM Teacher_Donated_Info;
DROP TABLE Teacher_Donated_Info;
--------------------
INSERT INTO Teacher_Donated_Info VALUES(1,'02-05-2019','A')
INSERT INTO Teacher_Donated_Info VALUES(2,'02-05-2019','A')

--------------------
```

Messages

```
9:17:17 PM     Started executing query at Line 289
               (1 row affected)
               (1 row affected)
               Total execution time: 00:00:00.045
```

```sql
--------------------
CREATE TABLE Blood_Bag_Quantity(
    Hospital_Code INT NOT NULL,
    A_Type_Fulfilled INT,
    B_Type_Fulfilled INT,
    AB_Type_Fulfilled INT,
    O_Type_Fulfilled INT,
    PRIMARY KEY(Hospital_Code),
    FOREIGN KEY(Hospital_Code) REFERENCES Hospital_Info(Hospital_Code)
);
--------------------
INSERT INTO Blood_Bag_Quantity VALUES(1,50,50,50,50);
--------------------

--------------------
SELECT * FROM Blood_Bag_Quantity;
drop TABLE Blood_Bag_Quantity;
--------------------
```

Results | Messages

	Hospital_Code	A_Type_Fulfilled	B_Type_Fulfilled	AB_Type_Fulfilled	O_Type_Fulfilled
1	1	33	48	50	40

Operations On The Database

- **Donated_Students Stored Procedure**: It be executed whenever we need to know the first and last name of the donated students and the name of the school they are in.
- **Donated_Teachers Stored Procedure:** It be executed whenever we need to know the first and last name of the donated Teachers and the name of the school they are in.

```sql
CREATE PROCEDURE Donated_Students
AS
SELECT Student_Donor.Student_FName , Student_Donor.Student_LName, School.School_Name
FROM Student_Donor, School, Student_Donated_Info
WHERE Student_Donor.School_Code = School.School_Code
AND Student_Donated_Info.Student_Code = Student_Donor.Student_Code

-------------------------
EXEC Donated_Students;
-------------------------
```

Results Messages

	Student_FName	Student_LName	School_Name
1	Sandy …	Tawfik …	North Central High School …
2	Andy …	Johnson …	Westfield High School …

```sql
CREATE PROCEDURE Donated_Teachers
AS
SELECT Teacher_Donor.Teacher_FName , Teacher_Donor.Teacher_LName, School.School_Name
FROM Teacher_Donor, School, Teacher_Donated_Info
WHERE Teacher_Donor.School_Code = School.School_Code
AND Teacher_Donated_Info.Teacher_Code = Teacher_Donor.Teacher_Code
----------------------------
EXEC Donated_Teachers;
--------------------------
```

Results Messages

	Teacher_FName	Teacher_LName	School_Name
1	Kelly …	Van Busum …	Carmel High School …
2	John …	Harris …	North Central High School …

Java

- Connection.java: Will have a public static Connection getConnection() function to make the connection between the database and java.

- It will also have a public static void runQuery(Connection conn, String query) that will take the Query and execute it and display its contents on the console

Stored Procedure used to extract data

```
49    public void start() throws SQLException {
50        Connection conn = getConnection();
51
52        while (true)
53        {
54             System.out.println("Show Donated Students (1)");
55             System.out.println("Show Donated Teachers(2)");
56             System.out.println("Show Student on file(3)");
57             System.out.println("Add New Donation(4)");
58             System.out.println("Exit(5)");
59
60           int selection = scanner.nextInt();
61
62           switch(selection){
63                case 1:
64                   Hospital.runQuery(conn, "EXEC Donated_Students;");
65                   System.out.println("    ");
66                   break;
67
68                case 2:
69                   Hospital.runQuery(conn, "EXEC Donated_Teachers;");
70                   System.out.println("    ");
71                   break;
```

Problems | Javadoc | Declaration | Console | Error Log

Main (13) [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (May 4, 2021, 9:41:09 PM)

```
1
Student_FName Sandy                    , Student_LName Tawfik                    , School_Name North Central High School
Student_FName Andy                     , Student_LName Johnson                   , School_Name Westfield High School

Show Donated Students (1)
Show Donated Teachers(2)
Show Student on file(3)
Add New Donation(4)
Exit(5)
```

```java
    public static void runQuery(Connection conn, String query) throws SQLException {
        //query = "SELECT *FROM Student_Donated_Info;";
        PreparedStatement readStatement = conn.prepareStatement(query);
        ResultSet resultSet = readStatement.executeQuery();

        //print results
        ResultSetMetaData rsmd = resultSet.getMetaData();
        int columnsNumber = rsmd.getColumnCount();
        while (resultSet.next()) {
            for (int i = 1; i <= columnsNumber; i++) {
                if (i > 1) System.out.print(", ");
                String columnValue = resultSet.getString(i);
                System.out.print(rsmd.getColumnName(i) + " " + columnValue);
            }
            System.out.println("");
        }
    }
}

```

```java
    public static Connection getConnection() throws SQLException {

        // configure this information
        String databaseName = "FINAL52";
        String user = "52Kah52";
        String password = "CSCI44300-Tesbiq7";

        String url = "jdbc:sqlserver://IN-CSCI-MSSQL2.ADS.IU.EDU\\SQL2019DEV:11433;database=" + databaseName + ";";

```

Problems | Javadoc | Declaration | Console | Error Log

JDBCTest [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Apr 25, 2021, 9:23:05 PM)

```
url: jdbc:sqlserver://IN-CSCI-MSSQL2.ADS.IU.EDU\SQL2019DEV:11433;database=FINAL52;
Status : Connected.
Show Donated Students (1)
Show Donated Teachers(2)
Show Student on file(3)
Add New Donation(4)
Exit(5)
```

```java
18    public static Connection getConnection() throws SQLException {
19
20        // configure this information
21        String databaseName = "FINAL52";
22        String user = "52Kah52";
23        String password = "CSCI44300-Tesbiq7";
24
25        String url = "jdbc:sqlserver://IN-CSCI-MSSQL2.ADS.IU.EDU\\SQL2019DEV:11433;database=" + databaseName + ";";
26
```

Problems | Javadoc | Declaration | Console | Error Log

DBCTest [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Apr 25, 2021, 9:23:05 PM)

```
Status : Connected.
Show Donated Students (1)
Show Donated Teachers(2)
Show Student on file(3)
Add New Donation(4)
Exit(5)
4
Add a student donor (1)
Add a teacher donor (2)
```

```java
    public static Connection getConnection() throws SQLException {

        // configure this information
        String databaseName = "FINAL52";
        String user = "52Kah52";
        String password = "CSCI44300-Tesbiq7";

        String url = "jdbc:sqlserver://IN-CSCI-MSSQL2.ADS.IU.EDU\\SQL2019DEV:11433;database=" + databaseName + ";";

```

Problems | Javadoc | Declaration | Console | Error Log

JDBCTest [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Apr 25, 2021, 9:23:05 PM)

```
4
Add a student donor (1)
Add a teacher donor (2)
1
Enter Student First Name
nour
Enter Student Last Name
kahile
Enter Student Grade (9th, 10th, 11th, 12th)
10th
```

```java
18    public static Connection getConnection() throws SQLException {
19
20        // configure this information
21        String databaseName = "FINAL52";
22        String user = "52Kah52";
23        String password = "CSCI44300-Tesbiq7";
24
25        String url = "jdbc:sqlserver://IN-CSCI-MSSQL2.ADS.IU.EDU\\SQL2019DEV:11433;database=" + databaseName + ";";
26
```

Problems | Javadoc | Declaration | Console | Error Log

JDBCTest [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Apr 25, 2021, 9:23:05 PM)

```
10th
Enter Student Age
15
Enter School Code
1
url: jdbc:sqlserver://IN-CSCI-MSSQL2.ADS.IU.EDU\SQL2019DEV:11433;database=FINAL52;
Status : Connected.
Show Donated Students (1)
Show Donated Teachers(2)
Show Student on file(3)
Add New Donation(4)
```

```sql
--kah52kah--
CREATE TABLE Student_Donor(
    Student_Code INT IDENTITY(1,1) NOT NULL,
    Student_FName CHAR(30),
    Student_LName CHAR(30),
    Student_Grade CHAR(4),
    Student_Age INT NOT NULL,
    School_Code INT NOT NULL,
    PRIMARY KEY(Student_Code),
    FOREIGN KEY(School_Code) REFERENCES School_Info(School_Code)
);
--------------------
SELECT * FROM Student_Donor;
DROP TABLE Student_Donor;
--------------------
```

Results | Messages

	Student_Code	Student_FName	Student_LName	Student_Grade	Student_Age	School_Code
1	1	Sandy …	Tawfik …	11th	17	1
2	2	Andy …	Johnson …	12th	18	1
3	9	nour …	kahile …	10th	15	1