

**Totally not Vampires:
A Non-Profit Blood Donation Organization**

Outline:

This database will be used to organize the hypothetical *Totally not Vampires* network of blood donation clinics. Due to the nature of the medical field it is vital that all aspects of the organization are properly tracked and maintained between donors, blood, employees, and clinics. The database will keep track of donor information, number and type of blood specimens, clinic locations, and the employees needed to manage demand.

Donors can come to any of our clinics and donate one unit of their blood specimen. Donor information will be accurately recorded by clinic employees and entered into the database. It is vital to maintain accurate information as blood samples must be linked to a donor. Each clinic is self-sustaining and can test blood samples on site for blood type and purity. Each clinic has a staff of employees with specific roles needed to accomplish our organizational goals.

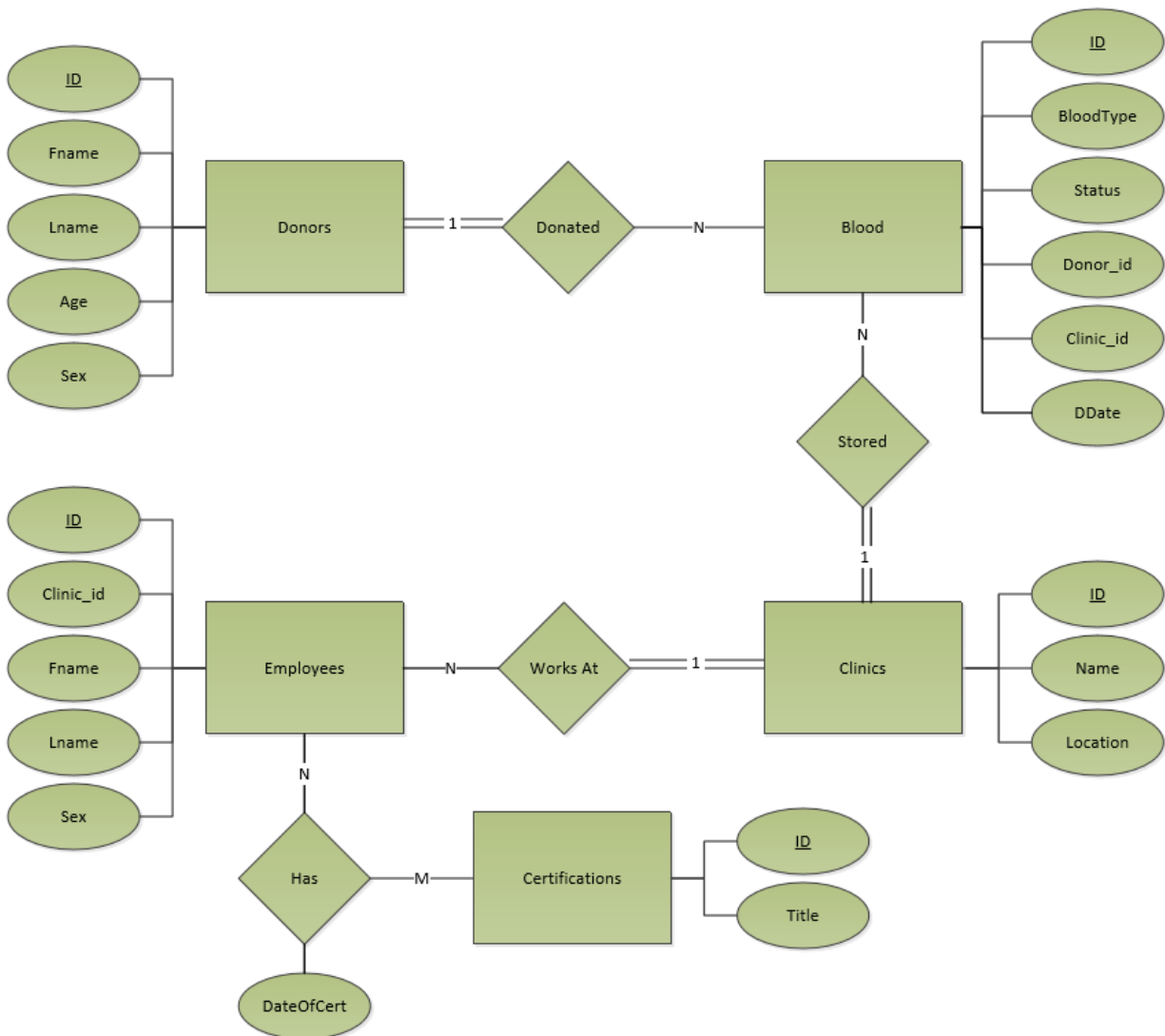
Database Outline:

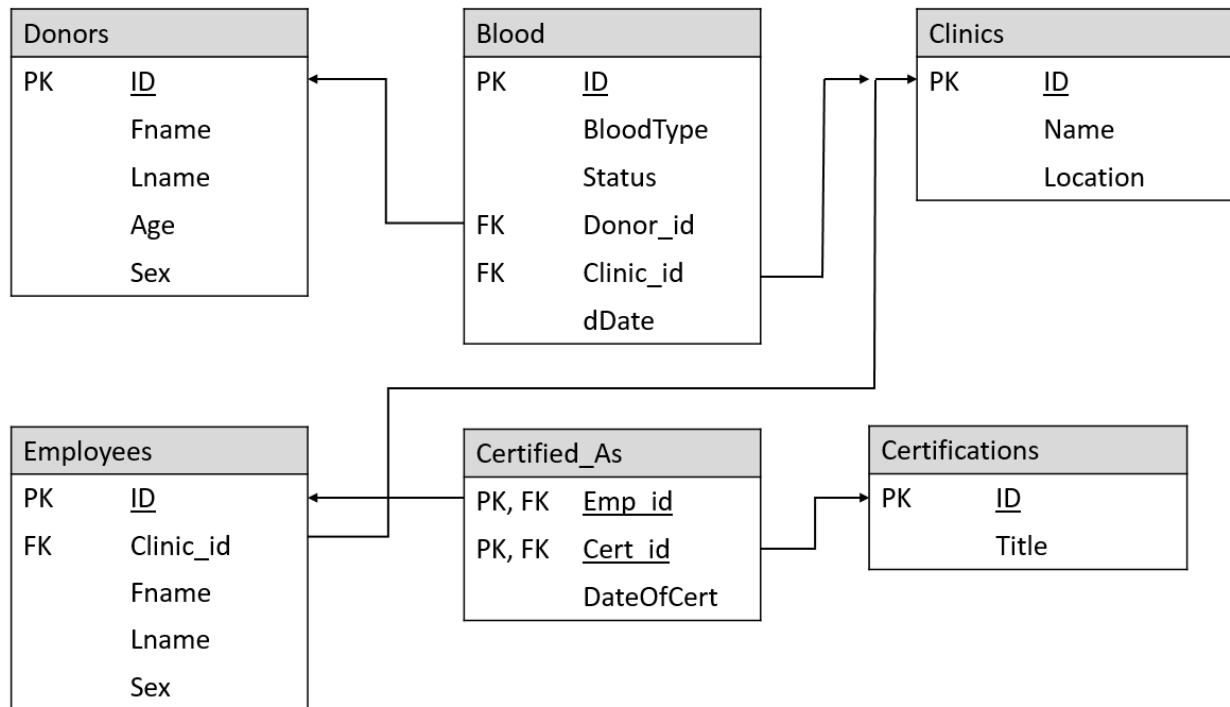
1. Donors:
 - a. Donors are tracked by a unique donor ID.
 - b. Personal information includes first name, last name, age, and sex.
 - i. The combination of the Donor's first and last name must be unique.
 - ii. Donor's sex is not a mandatory field. *
 - c. Donors must be at least 17 years of age.
 - d. Donors can donate at any clinic any number of times.
2. Blood:
 - e. Blood is tracked by a unique ID.
 - f. BloodType is how the blood is categorized: (A+/-, B+/-, AB+/-, O+/-, N/A).
 - i. N/A refers to blood that has not been processed yet.
 - g. Status refers to the state of the blood: (pure, N/A, impure).
 - i. N/A refers to blood that has not been processed yet.
 - h. Donor_id is a FK which references the ID of the Donor.
 - i. Blood specimens must have a donor
 - i. Clinic_id is a FK which references the ID of the Clinic that stores the blood.
 - i. Blood specimens must be stored at one clinic.
 - j. DonateDate is the date the blood sample was received.
3. Clinic:
 - a. Clinics are tracked by unique clinic ID.
 - b. Each clinic has a name and location.
 - i. The combination of the clinic's name and location must be unique
4. Employees:
 - a. Employees are tracked by unique Employee ID.
 - b. Clinic_id is a FK which references the Clinic ID which they work at.
 - i. Employees must work at a single Clinic.
 - c. Personal information includes Fname, Lname, and sex.

- i. The combination of an employee's first and last name must be unique
- 5. Certifications:
 - a. Certifications are tracked by a unique Certification ID.
 - b. Title is the type of certification: (Supervisor, phlebotomist, volunteer).
 - i. Each title must be unique.
- 6. Certified_As:
 - a. Many-to-many relationship between Employees and their Certifications
 - i. Emp_id is a FK which references Employee ID
 - ii. Cert_id is a FK which references Certifications ID
 - b. DateOfCert is the date which the certification was acquired.

***All fields have a NOT NULL constraint unless otherwise specified.**

ER Diagram of Database:



Database Schema:**MYSQL Queries:****-- TABLE CREATION QUERIES**

```

CREATE TABLE donors (
    id INT NOT NULL AUTO_INCREMENT,
    fname VARCHAR(50) NOT NULL,
    lname VARCHAR(50) NOT NULL,
    age INT NOT NULL,
    sex VARCHAR(15),
    UNIQUE `fullname` (fname, lname),
    PRIMARY KEY (id),
    CHECK (age > 17)
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
  
```

```

CREATE TABLE clinics (
    id INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(50) NOT NULL,
    location VARCHAR(50) NOT NULL,
    UNIQUE `name-location` (name, location),
    PRIMARY KEY (id)
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
  
```

```
CREATE TABLE blood (  
    id INT NOT NULL AUTO_INCREMENT,  
    bloodtype VARCHAR(10) NOT NULL,  
    status VARCHAR(50) NOT NULL,  
    donor_id INT NOT NULL,  
    clinic_id INT NOT NULL,  
    ddate DATE NOT NULL,  
    PRIMARY KEY (id),  
    FOREIGN KEY (donor_id) REFERENCES donors(id) ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (clinic_id) REFERENCES clinics(id) ON DELETE CASCADE ON UPDATE CASCADE  
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE employees (  
    id INT NOT NULL AUTO_INCREMENT,  
    clinic_id INT NOT NULL,  
    fname VARCHAR(50) NOT NULL,  
    lname VARCHAR(50) NOT NULL,  
    sex VARCHAR(15) NOT NULL,  
    UNIQUE `fullname` (fname, lname),  
    PRIMARY KEY (id),  
    FOREIGN KEY (clinic_id) REFERENCES clinics(id) ON DELETE CASCADE ON UPDATE CASCADE  
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE certifications (  
    id INT NOT NULL AUTO_INCREMENT,  
    title VARCHAR(50) NOT NULL,  
    UNIQUE (title),  
    PRIMARY KEY (id)  
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE certified_as (  
    emp_id INT NOT NULL,  
    cert_id INT NOT NULL,  
    DateOfCert DATE NOT NULL,  
    PRIMARY KEY (emp_id, cert_id),  
    FOREIGN KEY (emp_id) REFERENCES employees(id) ON DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (cert_id) REFERENCES certifications(id) ON DELETE CASCADE ON UPDATE CASCADE  
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

-- GENERAL USE QUERIES

-- INSERT DATA

```
INSERT INTO blood (bloodtype, status, donor_id, clinic_id, ddate) VALUES ([bloodtype], [status], [donor_id],  
[clinic_id], [ddate]);  
INSERT INTO certifications (title) VALUES ([title]);  
INSERT INTO clinics (name, location) VALUES ([name], [location]);  
INSERT INTO donors (fname, lname, age, sex) VALUES ([fname], [lname], [age], [sex]);  
INSERT INTO certified_as (emp_id, cert_id, dateOfCert) VALUES ([emp_id], [cert_id], [dateOfCert]);  
INSERT INTO employees (clinic_id, fname, lname, sex) VALUES ([clinic_id], [fname], [lname], [sex]);
```

-- SELECT DATA

```

SELECT b.id, bloodtype, status, donor_id, fname, lname, clinic_id, name, ddate
      FROM blood AS b
      INNER JOIN donors AS d ON b.donor_id=d.id
      INNER JOIN clinics AS c ON b.clinic_id=c.id
      ORDER BY d.id;
SELECT id, title FROM certifications;
SELECT id, name, location FROM clinics;
SELECT id, fname, lname, age, sex FROM donors;
SELECT e.id, e.fname, e.lname, cert.id, cert.title, ca.dateOfCert, c.id, c.name
      FROM employees AS e
      INNER JOIN certified_as AS ca ON e.id=ca.emp_id
      INNER JOIN certifications AS cert ON ca.cert_id=cert.id
      INNER JOIN clinics AS c ON e.clinic_id=c.id
      ORDER BY e.id;
SELECT e.id, e.fname, e.lname, e.sex, c.id, c.name FROM employees AS e
      INNER JOIN clinics AS c ON e.clinic_id=c.id
      ORDER BY e.id;

```

-- FILTER DATA

-----BLOOD DB-----

-- select by blood type

```

SELECT b.id, bloodtype, status, donor_id, fname, lname, clinic_id, name, ddate
      FROM blood AS b
      INNER JOIN donors AS d ON b.donor_id=d.id
      INNER JOIN clinics AS c ON b.clinic_id=c.id
      WHERE bloodtype=[bloodtype];

```

-- select by blood status

```

SELECT b.id, bloodtype, status, donor_id, fname, lname, clinic_id, name, ddate
      FROM blood AS b
      INNER JOIN donors AS d ON b.donor_id=d.id
      INNER JOIN clinics AS c ON b.clinic_id=c.id
      WHERE status=[status];

```

-- select by donor

```

SELECT b.id, bloodtype, status, donor_id, fname, lname, clinic_id, name, ddate
      FROM blood AS b
      INNER JOIN donors AS d ON b.donor_id=d.id
      INNER JOIN clinics AS c ON b.clinic_id=c.id
      WHERE b.donor_id=[donor_id];

```

-- select by clinic

```

SELECT b.id, bloodtype, status, donor_id, fname, lname, clinic_id, name, ddate
      FROM blood AS b
      INNER JOIN donors AS d ON b.donor_id=d.id
      INNER JOIN clinics AS c ON b.clinic_id=c.id
      WHERE c.id=[clinic_id];

```

-- select by date

```

SELECT b.id, bloodtype, status, donor_id, fname, lname, clinic_id, name, ddate
      FROM blood AS b
      INNER JOIN donors AS d ON b.donor_id=d.id

```

```
INNER JOIN clinics AS c ON b.clinic_id=c.id
WHERE ddate ['>' | '<' | '='] [user_date];
```

-- select by Aggregate COUNT

```
SELECT bloodtype, COUNT(bloodtype) AS 'Total'
      FROM blood AS b1
      GROUP BY bloodtype;
```

-----Clinic DB-----

-- select by location

```
SELECT id, name, location FROM clinics WHERE location=[location];
```

-----Donor DB-----

```
SELECT * FROM donors WHERE sex = [sex];
SELECT * FROM donors WHERE age ['>' | '<' | '='] [age];
```

-----Certified_as DB-----

-- select by employee id

```
SELECT e.id, e.fname, e.lname, cert.id, cert.title, ca.dateOfCert, c.id, c.name
      FROM employees AS e
      INNER JOIN certified_as AS ca ON e.id=ca.emp_id
      INNER JOIN certifications AS cert ON ca.cert_id=cert.id
      INNER JOIN clinics AS c ON e.clinic_id=c.id
      WHERE e.id=[employee_id]
      ORDER BY e.id;
```

-- select by certification title

```
SELECT e.id, e.fname, e.lname, cert.id, cert.title, ca.dateOfCert, c.id, c.name
      FROM employees AS e
      INNER JOIN certified_as AS ca ON e.id=ca.emp_id
      INNER JOIN certifications AS cert ON ca.cert_id=cert.id
      INNER JOIN clinics AS c ON e.clinic_id=c.id
      WHERE cert.id=[cert_id]
      ORDER BY e.id;
```

-- select by clinic id

```
SELECT e.id, e.fname, e.lname, cert.id, cert.title, ca.dateOfCert, c.id, c.name
      FROM employees AS e
      INNER JOIN certified_as AS ca ON e.id=ca.emp_id
      INNER JOIN certifications AS cert ON ca.cert_id=cert.id
      INNER JOIN clinics AS c ON e.clinic_id=c.id
      WHERE c.id=[clinic_id]
      ORDER BY e.id;
```

-- select by employees with NO certification

```
SELECT e1.id, e1.fname, e1.lname, c1.id, c1.name FROM employees AS e1
      INNER JOIN clinics AS c1 ON e1.clinic_id=c1.id
      WHERE e1.id
      NOT IN
      (SELECT e.id FROM employees AS e
      INNER JOIN certified_as AS ca ON e.id=ca.emp_id
      INNER JOIN certifications AS cert ON ca.cert_id=cert.id
      INNER JOIN clinics AS c ON e.clinic_id=c.id);
```

-----Employee DB-----

-- select by sex

```
SELECT e.id, e.fname, e.lname, e.sex, c.id, c.name FROM employees AS e
      INNER JOIN clinics AS c ON e.clinic_id=c.id
      WHERE sex=[sex];
```

-- select by age

```
SELECT e.id, e.fname, e.lname, e.sex, c.id, c.name FROM employees AS e
      INNER JOIN clinics AS c ON e.clinic_id=c.id
      WHERE e.clinic_id=[clinic_id];
```

-- DELETE DATA

```
DELETE FROM donors WHERE id=[donor_id];
DELETE FROM blood WHERE id=[blood_id];
DELETE FROM certifications WHERE id=[cert_id];
DELETE FROM clinics WHERE id=[clinic_id];
DELETE FROM certified_as WHERE emp_id=[emp_id] AND cert_id=[cert_id];
DELETE FROM employees WHERE id=[emp_id];
```

-- MODIFY DATA

```
UPDATE blood SET
      bloodtype=[bloodtype],
      status =[status],
      donor_id =[donor_id],
      clinic_id =[clinic_id],
      ddate =[ddate]
      WHERE id=[id];
```

```
UPDATE certifications SET
      title =[title]
      WHERE id=[id];
```

```
UPDATE clinics SET
      name=[name],
      location=[location]
      WHERE id=[clinic_id];
```

```
UPDATE donors SET
```

```
    fname=[first_name],
```

```
    lname=[last_name],
```

```
    age=[age],
```

```
    sex=[sex]
```

```
WHERE id=[donor_id];
```

```
UPDATE certified_as SET
```

```
    emp_id=[emp_id],
```

```
    cert_id=[cert_id],
```

```
    DateOfCert=[DateOfCert]
```

```
WHERE emp_id=[emp_id_x] AND cert_id=[cert_id_x];
```

```
UPDATE employees SET
```

```
    fname=[fname],
```

```
    lname = [lname],
```

```
    clinic_id = [clinic_id],
```

```
    sex = [sex]
```

```
WHERE id=[id];
```

```
-- DROP ALL TABLES
```

```
SET foreign_key_checks = 0;
```

```
DROP TABLE certifications;
```

```
DROP TABLE certified_as;
```

```
DROP TABLE employees;
```

```
DROP TABLE clinics;
```

```
DROP TABLE blood;
```

```
DROP TABLE donors;
```

```
SET foreign_key_checks = 1;
```