## **1. Entities**

1. **Patients**
2. **Account**
3. **Doctor**
4. **Donors**
5. **Donation\_Event**
6. **Request**
7. **Blood\_Bank**
8. **Blood\_Inventory**

## **2. Relationships**

| **Relationship** | **Type** | **Description** |
| --- | --- | --- |
| Patients — create — Account | 1-to-1 | Each patient has one account |
| Doctor — create — Account | 1-to-1 | Each doctor has one account |
| Donors — enroll\_in — Donation\_Event | many-to-many | Many donors can join many donation events |
| Blood\_Bank — organize — Donation\_Event | 1-to-many | One blood bank can organize many events |
| Doctor — work\_at — Blood\_Bank | many-to-one | Many doctors can work at one blood bank |
| Blood\_Bank — store — Blood\_Inventory | 1-to-many | One blood bank stores many blood units |
| Patients/Donors — send — Request | many-to-one | Many patients or donors can send requests |
| Request — manage — Blood\_Bank | many-to-one | Requests are managed by a blood bank |

## **3. Tables**

### **Patients**

| **Field** | **Type** | **Key** | **Description** |
| --- | --- | --- | --- |
| PatientID | INT | PK | Unique ID |
| Name | VARCHAR |  | Patient’s name |
| Gender | CHAR |  | M/F |
| DOB | DATE |  | Birth date |
| Contact | VARCHAR |  | Phone/email |

**Doctor**

| **Field** | **Type** | **Key** | **Description** |
| --- | --- | --- | --- |
| DoctorID | INT | PK | Unique ID |
| Name | VARCHAR |  | Doctor name |
| Specialization | VARCHAR |  | Field of expertise |
| BankID | INT | FK | Works at Blood\_Bank |

### **Account**

| **Field** | **Type** | **Key** |
| --- | --- | --- |
| AccountID | INT | PK |
| Username | VARCHAR |  |
| Password | VARCHAR |  |
| Role | VARCHAR | (‘Patient’, ‘Doctor’, etc.) |
| PatientID | INT | FK (nullable) |
| DoctorID | INT | FK (nullable) |

### **Donors**

| **Field** | **Type** | **Key** |
| --- | --- | --- |
| DonorID | INT | PK |
| Name | VARCHAR |  |
| BloodType | VARCHAR |  |
| Contact | VARCHAR |  |
| Address | VARCHAR |  |

### **Donation\_Event**

| **Field** | **Type** | **Key** |
| --- | --- | --- |
| EventID | INT | PK |
| BankID | INT | FK (organized by Blood\_Bank) |
| EventDate | DATE |  |
| Location | VARCHAR |  |

**Blood\_Inventory**

| **Field** | **Type** | **Key** |
| --- | --- | --- |
| BloodID | INT | PK |
| BankID | INT | FK |
| BloodType | VARCHAR |  |
| Quantity | INT |  |
| ExpiryDate | DATE |  |

**Request**

| **Field** | **Type** | **Key** |
| --- | --- | --- |
| RequestID | INT | PK |
| PatientID | INT | FK (nullable) |
| DonorID | INT | FK (nullable) |
| BankID | INT | FK |
| RequestDate | DATE |  |
| Status | VARCHAR | (‘Pending’, ‘Approved’, etc.) |

**Blood\_Bank**

| **Field** | **Type** | **Key** |
| --- | --- | --- |
| BankID | INT | PK |
| BankName | VARCHAR |  |
| Location | VARCHAR |  |

## **4. Normalization**

| **Normal Form** | **Condition** | **Applied in your system** |
| --- | --- | --- |
| **1NF** | Each column holds atomic values (no lists, no repeating groups) | All fields store single values only |
| **2NF** | All non-key attributes depend on the full primary key | Each table has single-column PKs, so OK |
| **3NF** | No transitive dependency (non-key depends only on key) | Example: In Account, “Role” depends only on AccountID, not through another field |

**5 Summary for Report**

**3NF and Relationships Summary:** The Blood Donation Management System was normalized up to Third Normal Form (3NF) to ensure data consistency and eliminate redundancy. Each entity (Patients, Donors, Doctor, Blood Bank, etc.) has a unique primary key, and all non-key attributes depend only on the key.

The relationships between entities have been defined using foreign keys. For example, Doctors work at a Blood Bank, Donors enroll in Donation Events, and Patients send Requests managed by Blood Banks. This structure maintains data integrity and enables efficient data retrieval for managing blood donations and requests.