**BLOOD DONATION MANAGEMENT SYSTEM**

**PROJECT DOCUMENTATION**

Employee ID: **SP11235**

Employee Name: **Annu Sebastian**

Project Name: **BLOOD DONATION MANAGEMENT SYSTEM**

**Project Background:**

The Blood Donation Management System efficiently manages blood donations, donors, recipients, and inventory, ensuring a stable blood supply for healthcare institutions. The system's tables include donor and recipient information, transaction records, and medical histories, enhancing transparency and timely blood allocation. Staff and blood transfer modules further optimize resource allocation and minimize shortages.

**Project Objective:**

The Blood Donation Management System aims to optimize blood donation processes, ensuring a reliable blood supply. It streamlines donor and recipient management, facilitates efficient allocation, tracks inventory, and promotes transparency. Staff management and blood transfer modules further enhance operational efficiency.

**Activities:**

1. **Creating tables using Queries**

Created 13 tables with relevant fields for managing the blood donation system. Queries were automatically generated while I created the ER Diagram using dbdiagram.io online software.

**Donors Table:** Stores donor information, including personal details, contact information, and blood type, each assigned a unique DonorID.

**Donor Transaction Table:** Records individual donation transactions, capturing donor details, donation dates, and quantities donated.

**Recipient Table:** Manages recipient information, such as personal details, contact information, and blood type, with a unique RecipientID.

**Recipient Transaction Table:** Logs recipient requests and transactions, including request status and blood quantity requested.

**Donation Center Table**: Stores data related to donation centers, including center details and services offered.

**Blood Type Table:** Contains information about various blood types, including BloodTypeID, blood group, Rh factor, and descriptions.

**Blood Stock Table:** Tracks available blood stock, updated continuously to reflect changes in stock levels.

**Staff Table:** Manages staff working at donation centers, with staff details and unique StaffID.

**Blood Request Table:** Records recipient blood requests, including request details and statuses.

**Blood Transfer Table:** Manages blood transfers between centers, capturing transfer details and quantities transferred.

**Medical History Table:** Stores donor medical history, including conditions, medications, allergies, and chronic conditions.

Create Table queries are provided below

CREATE TABLE [DONORS] (

[DonorID] INT PRIMARY KEY,

[FirstName] VARCHAR(100),

[LastName] VARCHAR(100),

[DateOfBirth] DATE,

[ContactNumber] VARCHAR(100),

[BloodTypeID] INT

)

GO

CREATE TABLE [DONOR\_TRANSACTION] (

[DTransactionID] INT PRIMARY KEY,

[DonorID] INT,

[DonationDate] DATE,

[DCenterID] INT,

[DonationStatus] VARCHAR(100),

[DQuantity] INT

)

GO

CREATE TABLE [BLOOD\_TYPE] (

[BloodTypeID] INT PRIMARY KEY,

[BloodGroup] VARCHAR(10),

[RhFactor] VARCHAR(100),

[BloodTypeDescription] VARCHAR(100)

)

GO

CREATE TABLE [RECIPIENTS] (

[RecipientID] INT Primary Key,

[FirstName] VARCHAR(100),

[LastName] VARCHAR(100),

[DateOfBirth] DATE,

[ContactNumber] VARCHAR(100),

[BloodTypeID] INT

)

GO

CREATE TABLE [RECIPIENT\_TRANSACTIONS] (

[RTransactionID] INT PRIMARY KEY,

[RecipientID] INT,

[RequestDate] DATE,

[RequestStatus] VARCHAR(100),

[DCenterID] INT,

[RQuantity] INT

)

GO

CREATE TABLE [DONATION\_CARD] (

[CardID] INT PRIMARY KEY,

[DonorID] INT,

[DonorTransactionID] INT,

[BloodTypeID] INT,

[ExpirationDate] DATE

)

GO

CREATE TABLE [DONATION\_CENTRE] (

[DCenterID] INT PRIMARY KEY,

[CenterName] VARCHAR(100),

[Address] VARCHAR(100),

[ContactNumber] VARCHAR(100),

[OperatingHours] INT,

[ContactPerson] VARCHAR(100),

[ServicesOffered] VARCHAR(100)

)

GO

CREATE TABLE [BLOOD\_STOCK] (

BloodStockID INT IDENTITY(1,1) PRIMARY KEY,

BloodTypeID INT,

TransactionDate DATE,

TotalQuantity INT

)

GO

CREATE TABLE [BLOOD\_INVENTORY] (

[BloodInventoryID] INT PRIMARY KEY,

[BloodTypeID] INT,

[DonationDate] DATE,

[ExpirationDate] DATE,

[DCenterID] INT,

[DTransactionID] INT,

[Quantity] INT

)

GO

CREATE TABLE [MEDICAL\_HISTORY] (

[MedicalHistoryID] INT Primary Key,

[DonorID] INT,

[MedicalCondition] VARCHAR(100),

[Medications] VARCHAR(100),

[Allergies] VARCHAR(100),

[ChronicConditions] VARCHAR(100)

)

GO

CREATE TABLE [STAFF] (

[StaffID] INT Primary Key,

[FirstName] VARCHAR(100),

[LastName] VARCHAR(100),

[Position] VARCHAR(100),

[ContactNumber] INT,

[DCenterID] INT

)

GO

CREATE TABLE [BLOOD\_REQUEST] (

[RequestID] INT Primary Key,

[RecipientID] INT,

[RequestDate] DATE,

[RequestedBloodType] INT,

[RequestStatus] VARCHAR(100),

[DCenterID] INT

)

GO

CREATE TABLE [BLOOD\_TRANSFER] (

[TransferID] INT Primary Key,

[SourceDonationCenterID] INT,

[DestinationDonationCenterID] INT,

[BloodInventoryID] INT,

[TransferDate] DATE,

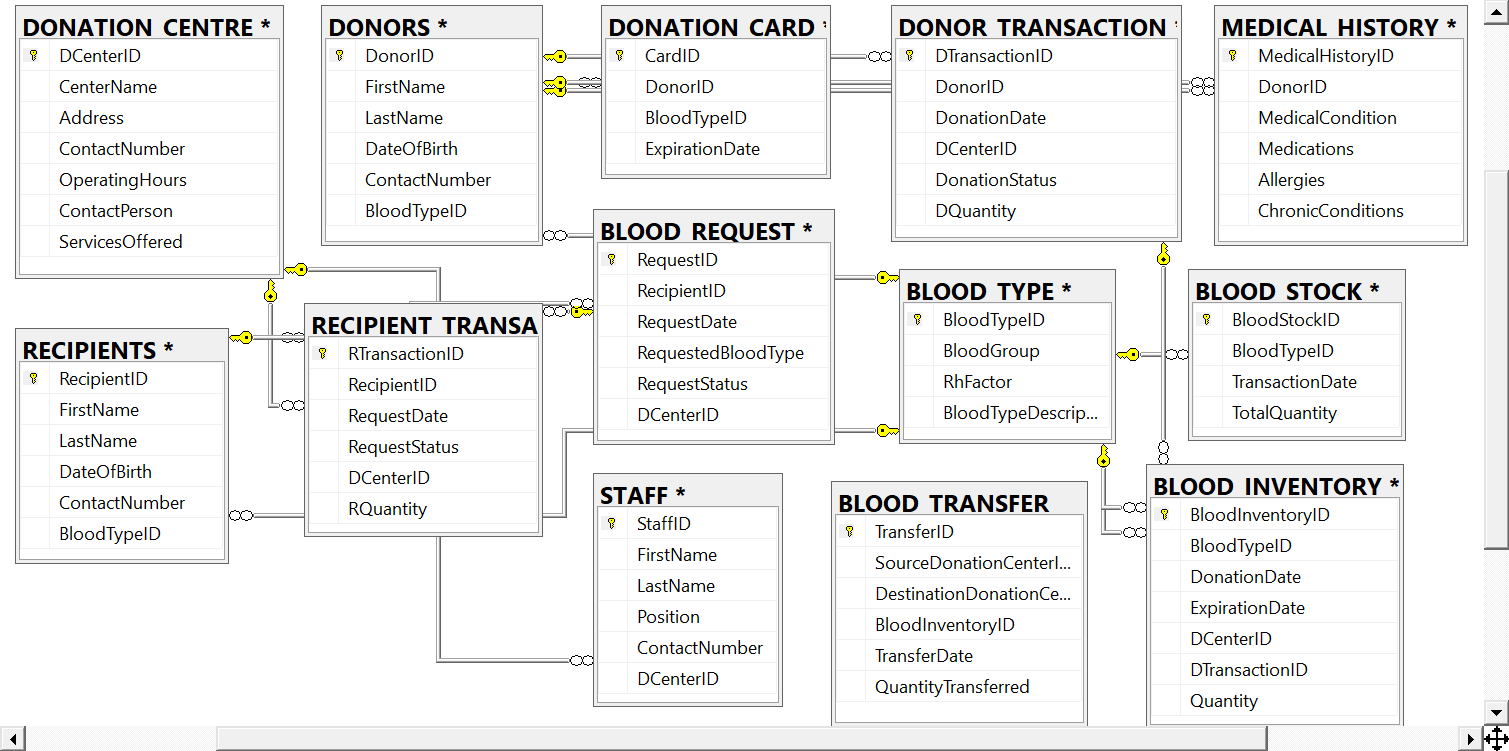
[QuantityTransferred] INT

)

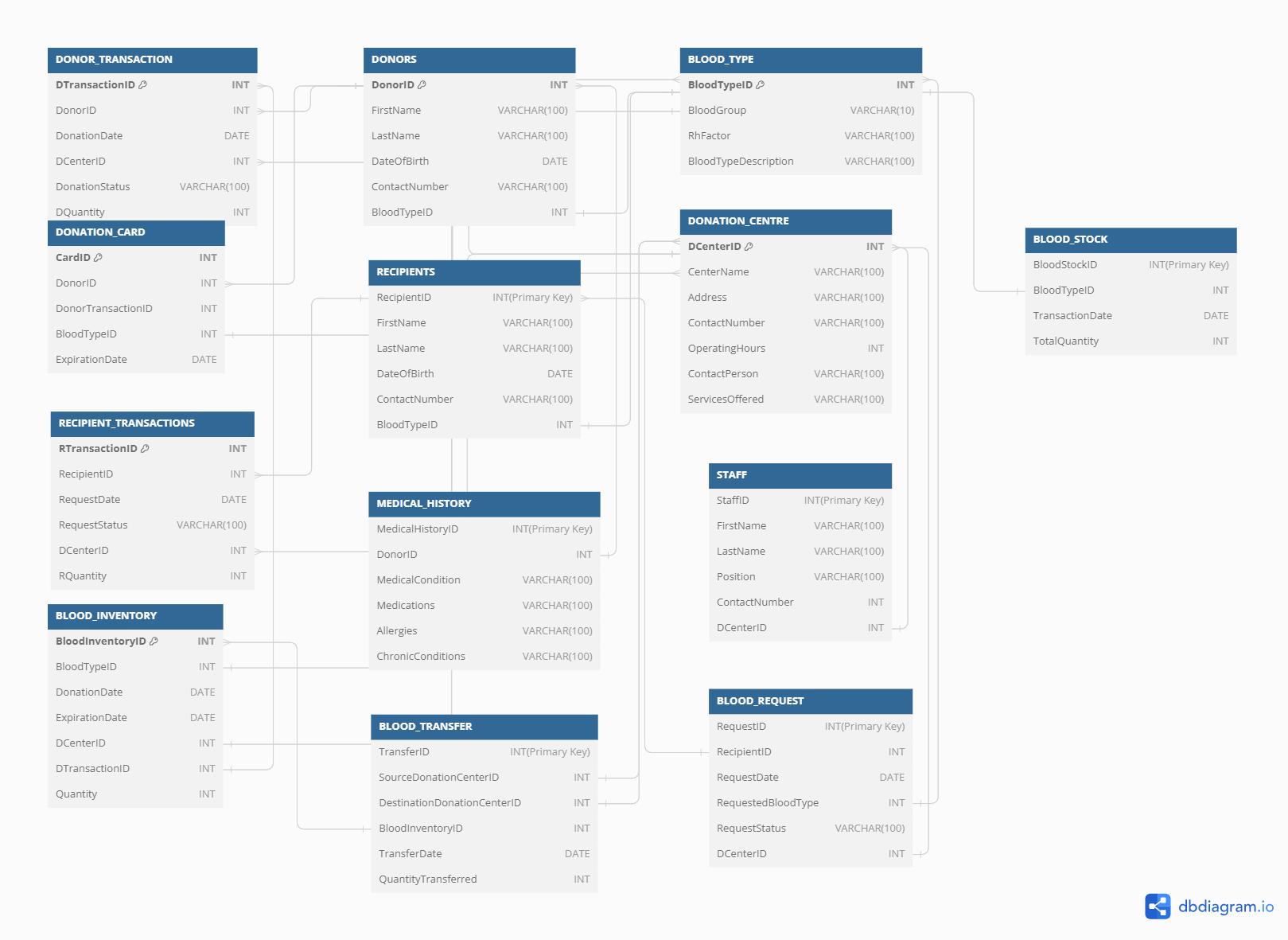
GO

1. **Schema/Diagram**

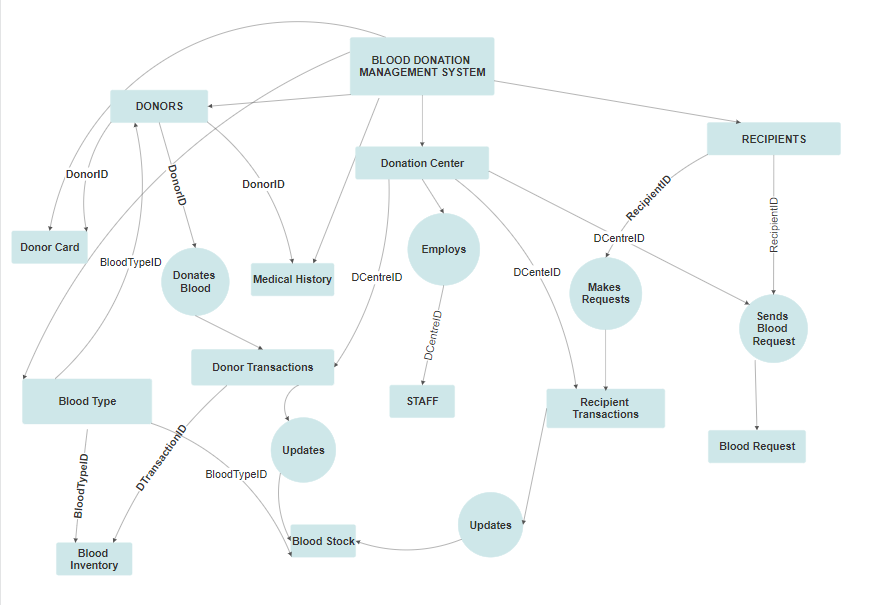
**Database Diagram**

****

**ER Diagram**

****

**Dataflow Diagram**

****

1. **Entering Data in tables using queries and triggers**

Entered data into the following tables manually **using SQL queries.**

1. BLOOD\_TYPE TABLE - 8 Records

INSERT INTO Blood\_Type (BloodTypeID, BloodGroup, RhFactor, BloodTypeDescription)

VALUES (1, 'A', '+', 'A positive - Common blood type');

1. DONORS TABLE – 20 Records

INSERT INTO DONORS (DonorID, FirstName, LastName, DateOfBirth, ContactNumber, BloodTypeID)

VALUES (1, 'John', 'Smith', '1985-05-10', '+123456789001', 1); -- Blood Type A+

1. RECIPIENTS TABLE – 24 Records

INSERT INTO RECIPIENTS (RecipientID, FirstName, LastName, DateOfBirth, ContactNumber, BloodTypeID)

VALUES (1, 'David', 'Jones', '1992-03-18', '+123456789101', 1), -- Blood Type A+

1. DONATION CENTRE TABLE – 10 Records

INSERT INTO DONATION\_CENTRE (DCenterID, CenterName, Address, ContactNumber, OperatingHours, ContactPerson, ServicesOffered)

VALUES (1, 'City Hospital Blood Center', '123 Main St, City, Country', '+1234567890', 8, 'John Doe', 'Blood donation, medical checkups');

1. RECIPIENT TRANSACTION TABLE – 26 Records

INSERT INTO RECIPIENT\_TRANSACTIONS (RTransactionID,RecipientID, RequestDate, RequestStatus, DCenterID, RQuantity)

VALUES (1,1, '2023-10-01', 'Successful', 1, 200);

1. DONOR TRANSACTION TABLE – 30 Records

INSERT INTO DONOR\_TRANSACTION (DTransactionID,DonorID, DonationDate, DCenterID, DonationStatus, DQuantity)

VALUES (1,1, '2023-10-01', 1, 'Successful', 350);

1. DONATION CARD TABLE – 20 Records

INSERT INTO DONATION\_CARD (CardID, DonorID, BloodTypeID, ExpirationDate)

VALUES (1, 1, 1, '2024-10-01');

1. MEDICAL HISTORY TABLE – 20 Records

INSERT INTO MEDICAL\_HISTORY (MedicalHistoryID, DonorID, MedicalCondition, Medications, Allergies, ChronicConditions)

VALUES (1, 1, 'Hypertension', 'Lisinopril', 'Pollen', 'None');

1. BLOOD INVENTORY TABLE – 30 Records

INSERT INTO BLOOD\_INVENTORY (BloodInventoryID, BloodTypeID, DonationDate, ExpirationDate, DCenterID, DTransactionID, Quantity)

VALUES (1, 1, '2023-10-01', '2024-01-01', 1, 1, 350);

1. STAFF TABLE – 10 Records

INSERT INTO STAFF (StaffID, FirstName, LastName, Position, ContactNumber, DCenterID)

VALUES (1, 'John', 'Smith', 'Nurse', '+1234567890', 1);

1. BLOOD REQUEST TABLE – 26 Records

INSERT INTO BLOOD\_REQUEST (RequestID, RecipientID, RequestDate, RequestedBloodType, RequestStatus, DCenterID)

VALUES (1, 1, '2023-10-01', 1, 'Successful', 1);

1. BLOOD TRANSFER TABLE – 10 Records

INSERT INTO BLOOD\_TRANSFER (TransferID, SourceDonationCenterID, DestinationDonationCenterID, BloodInventoryID, TransferDate, QuantityTransferred)

VALUES (1, 1, 2, 1, '2023-10-01', 100);

**USING TRIGGERS**

1. BLOOD STOCK TABLE

Records were inserted into the blood stock table using two triggers working on two different tables, Donor Transaction and Recipient Transactions.

The two triggers are fired after an insertion happens in both tables.

**1)INSERT Trigger for Donor Transactions:**

Create an AFTER INSERT trigger on the Donor Transaction table to update the BLOOD\_STOCK table when a new donor transaction is made. This trigger should add the donated quantity to the stock for the corresponding blood type.

**Working:**

CREATE TRIGGER trgOnInsert1 ON [dbo].[Donor\_Transaction]

After INSERT

AS

BEGIN

DECLARE @blood\_type\_id INT;

DECLARE @donation\_date DATE;

DECLARE @total\_quantity INT;

-- Get values from the inserted rows

select @blood\_type\_id = BloodTypeID from DONORS Inner join DONOR\_TRANSACTION ON DONORS.DonorID = DONOR\_TRANSACTION.DonorID

select @donation\_date = DonationDate from DONOR\_TRANSACTION

select @total\_quantity = DQuantity from DONOR\_TRANSACTION

IF NOT EXISTS (

SELECT 1

FROM BLOOD\_STOCK

WHERE BloodTypeID = @blood\_type\_id

AND TransactionDate = @donation\_date

)

BEGIN

-- Insert the initial data when the condition is not met

INSERT INTO BLOOD\_STOCK (BloodTypeID, TransactionDate, TotalQuantity)

VALUES (@blood\_type\_id, @donation\_date, @total\_quantity);

END

ELSE

BEGIN

UPDATE BLOOD\_STOCK

SET TotalQuantity = TotalQuantity + @total\_quantity

WHERE BloodTypeID = @blood\_type\_id

AND TransactionDate = @donation\_date;

END

PRINT 'AFTER INSERT trigger fired'

END

GO

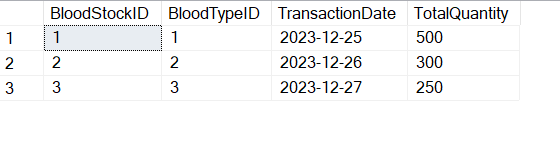
This Trigger was executed first

Next the following query was executed

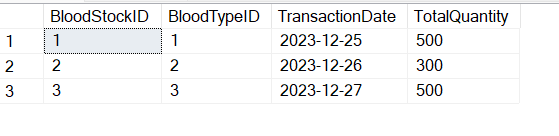
INSERT INTO DONOR\_TRANSACTION

VALUES (35, 3, '2023-12-27', 2, 'Successful', 250)

trgOnInsert1 gets fired after insertion on DONOR TRANSACTION TABLE and gets reflected on the BLOOD STOCK TABLE



Again an insert query is run on DONOR TRANSACTION TABLE on the same day and for the same blood type. So the Total Quantity of blood gets added up with the input quantity.



**2) INSERT Trigger for Recipient Transactions:**

Create an AFTER INSERT trigger on the Recipient Transaction table to update the BLOOD\_STOCK table when a new recipient transaction is made. This trigger should add the received quantity to the stock for the corresponding blood type.

**Working:**

CREATE TRIGGER trgOnInsert2 ON [dbo].[Recipient\_Transactions]

After INSERT

AS

BEGIN

DECLARE @blood\_type\_id INT;

DECLARE @request\_date DATE;

DECLARE @total\_quantity INT;

-- Get values from the inserted rows

select @blood\_type\_id = BloodTypeID from RECIPIENTS Inner join RECIPIENT\_TRANSACTIONS ON RECIPIENTS.RecipientID = RECIPIENT\_TRANSACTIONS.RecipientID

select @request\_date = RequestDate from RECIPIENT\_TRANSACTIONS

select @total\_quantity = RQuantity from RECIPIENT\_TRANSACTIONS

IF NOT EXISTS (

SELECT 1

FROM BLOOD\_STOCK

WHERE BloodTypeID = @blood\_type\_id

AND TransactionDate = @request\_date

)

BEGIN

-- Insert the initial data when the condition is not met

INSERT INTO BLOOD\_STOCK (BloodTypeID, TransactionDate, TotalQuantity)

VALUES (@blood\_type\_id, @request\_date, @total\_quantity);

END

ELSE

BEGIN

UPDATE BLOOD\_STOCK

SET TotalQuantity = TotalQuantity - @total\_quantity

WHERE BloodTypeID = @blood\_type\_id

AND TransactionDate = @request\_date;

END

PRINT 'AFTER INSERT trigger fired'

END

GO

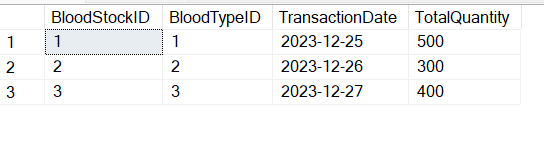
This Trigger was executed first

Next the following query was executed

INSERT INTO RECIPIENT\_TRANSACTIONS

VALUES (29, 3, '2023-12-27', 'Successful',2,100)

trgOnInsert2 gets fired after insertion on RECIPIENT TRANSACTION TABLE and gets reflected on the BLOOD STOCK TABLE



In the output the Total Quantity got subtracted with the input quantity as the transaction was made on the same date and for the same blood type as that of the last record in the BLOOD STOCK TABLE.

With these triggers in place, when a new donor transaction or recipient transaction is added to the respective tables, the BLOOD\_STOCK table will be updated to reflect the total quantity available for each blood type on that day.

**4. Creating Views**

1. VIEW 1 - Donor and Transaction Summary View:

This view combines information from the "Donors" and "Donor Transaction" tables.

It presents a summary of each donor's essential details (e.g., name, contact number, blood type) along with their donation history, including the number of donations and total quantity donated.

CREATE VIEW [dbo].[view1] AS

SELECT DONORS.DonorID, DONORS.FirstName, DONORS.LastName, DONORS.ContactNumber, COUNT(DONOR\_TRANSACTION.DonationStatus) AS NO\_OF\_DONATIONS,

SUM(DONOR\_TRANSACTION.DQuantity) AS DONATED\_QUANTITY FROM DONORS

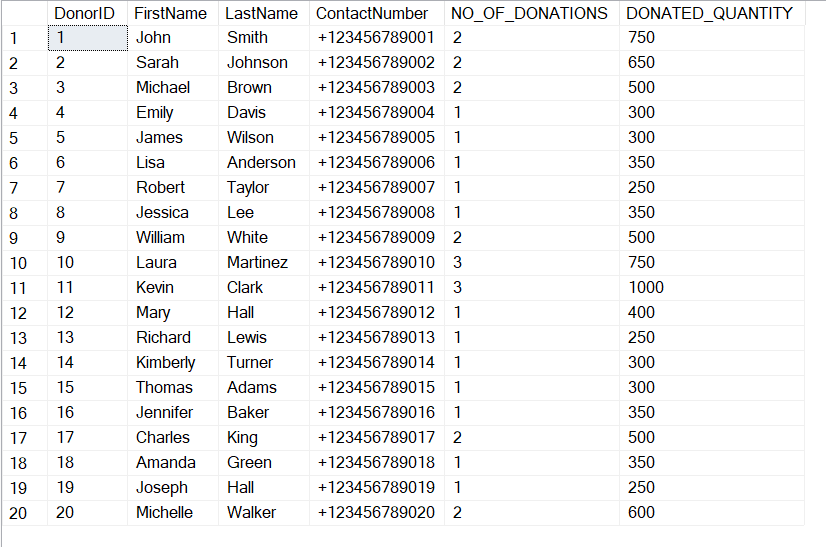
INNER JOIN DONOR\_TRANSACTION ON DONORS.DonorID = DONOR\_TRANSACTION.DonorID

GROUP BY DONORS.DonorID, DONORS.FirstName, DONORS.LastName, DONORS.ContactNumber, DONOR\_TRANSACTION.DonationStatus

HAVING DonationStatus = 'Successful'

GO

SELECT \* FROM view1



1. VIEW2 - Recipient Request Status View:

This view merges data from the "Recipients” and "Blood Request" tables.

It provides a comprehensive overview of recipient requests, showing details about the recipient, request date, requested blood type, and the current status of the request.

CREATE VIEW [dbo].[view2] AS

SELECT RECIPIENTS.RecipientID, RECIPIENTS.FirstName, RECIPIENTS.LastName, BLOOD\_REQUEST.RequestDate, BLOOD\_REQUEST.RequestedBloodType,

BLOOD\_REQUEST.RequestStatus FROM RECIPIENTS

INNER JOIN BLOOD\_REQUEST ON RECIPIENTS.RecipientID = BLOOD\_REQUEST.RecipientID

GO

SELECT \* FROM view2

1. VIEW3 - Blood Center Inventory Dashboard:

This view incorporates data from the “Blood Inventory” and "Blood Type" tables.

It offers a real-time dashboard displaying the current inventory status at each donation center, organized by blood type and Rh factor. This view helps staff manage blood stock efficiently.

CREATE VIEW [dbo].[view3] AS

SELECT BLOOD\_INVENTORY.DCenterID, BLOOD\_TYPE.BloodGroup, BLOOD\_TYPE.RhFactor, SUM(BLOOD\_INVENTORY.Quantity) AS AVILABLE\_QUANTITY

FROM BLOOD\_INVENTORY

INNER JOIN BLOOD\_TYPE ON BLOOD\_TYPE.BloodTypeID = BLOOD\_INVENTORY.BloodTypeID

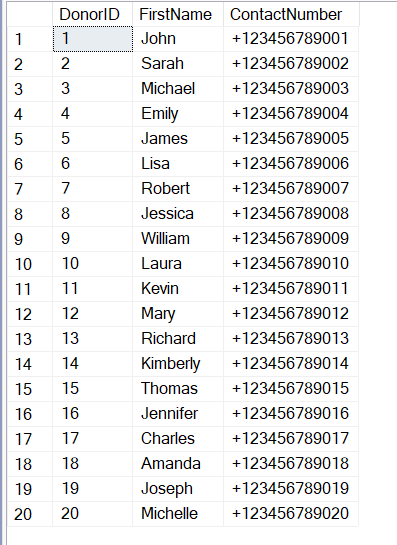
GROUP BY BLOOD\_INVENTORY.DCenterID, BLOOD\_TYPE.BloodGroup, BLOOD\_TYPE.RhFactor

GO

SELECT \* FROM view3

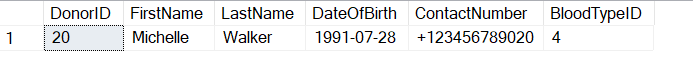
1. **Reports using queries**
2. Report on List all donors along with their contact numbers.

SELECT DonorID, FirstName, ContactNumber FROM DONORS



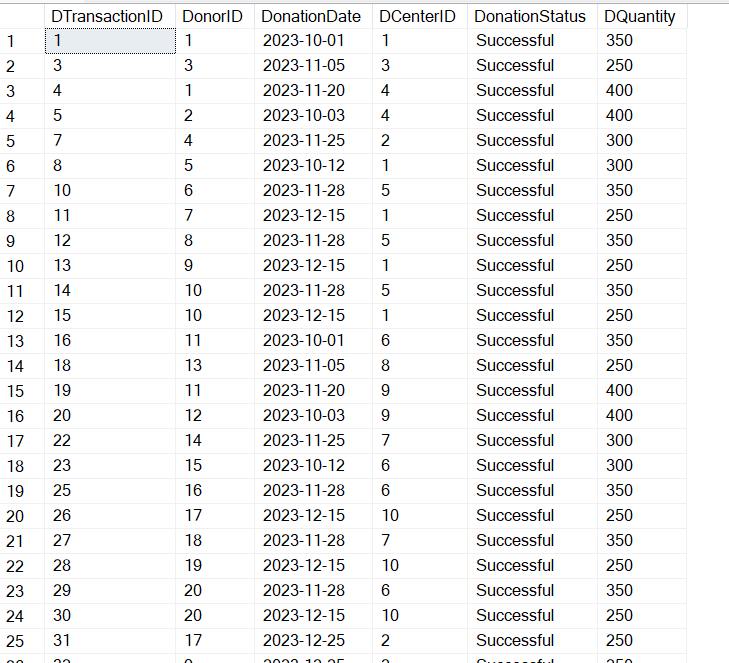
1. Report on details of a specific donor by DonorID.

SELECT \* FROM DONORS WHERE DonorID=20



1. Report on all successful donor transactions.

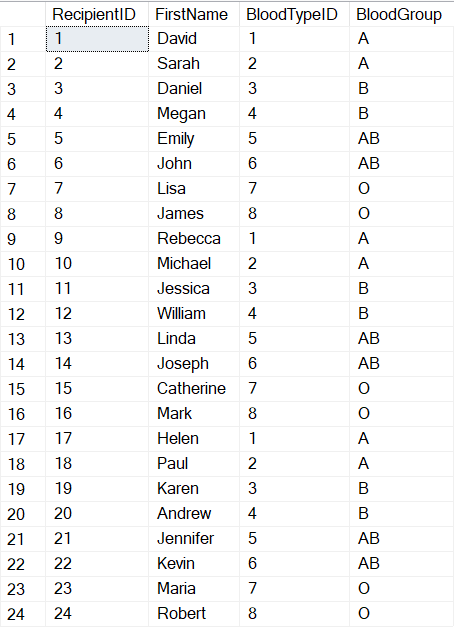
SELECT \* FROM DONOR\_TRANSACTION WHERE DonationStatus = 'Successful'



1. Report on all recipients with their respective blood types.

SELECT RECIPIENTS.RecipientID, RECIPIENTS.FirstName, RECIPIENTS.BloodTypeID, BLOOD\_TYPE.BloodGroup FROM RECIPIENTS

INNER JOIN BLOOD\_TYPE ON RECIPIENTS.BloodTypeID = BLOOD\_TYPE.BloodTypeID



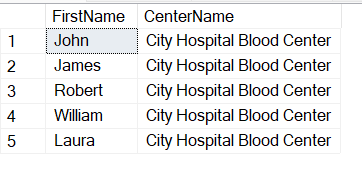
1. Report on the names of donors who donated at a specific donation center (e.g., DonationCenterID = 1).

SELECT DONORS.FirstName, DONATION\_CENTRE.CenterName FROM DONORS

INNER JOIN DONOR\_TRANSACTION ON DONORS.DonorID = DONOR\_TRANSACTION.DonorID

INNER JOIN DONATION\_CENTRE ON DONOR\_TRANSACTION.DCenterID = DONATION\_CENTRE.DCenterID

WHERE DONATION\_CENTRE.DCenterID = 1



1. Report on all donation transactions, including donor and blood type information.

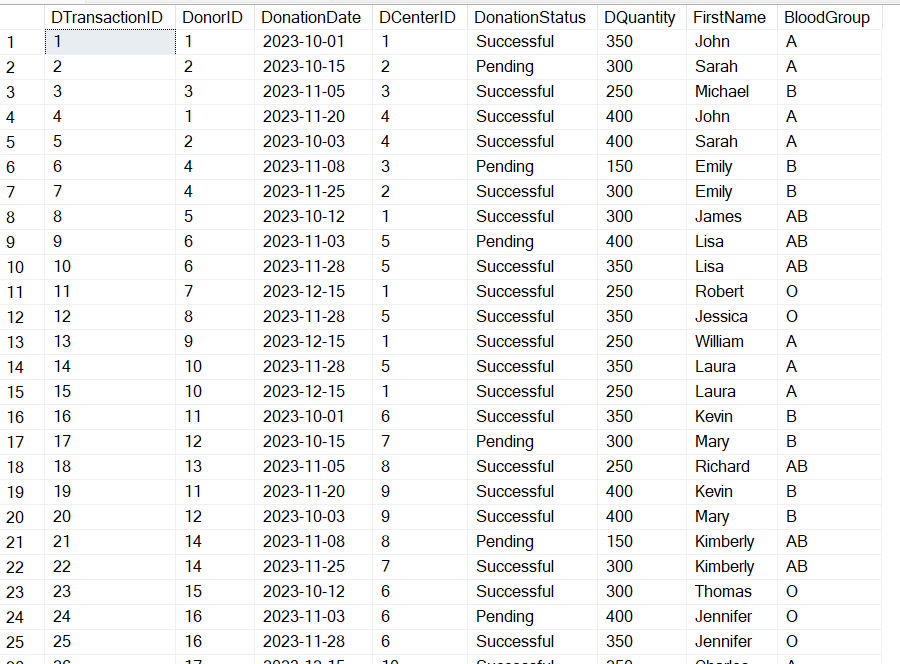
SELECT DONOR\_TRANSACTION.DTransactionID, DONOR\_TRANSACTION.DonorID, DONOR\_TRANSACTION.DonationDate, DONOR\_TRANSACTION.DCenterID,

DONOR\_TRANSACTION.DonationStatus, DONOR\_TRANSACTION.DQuantity,

DONORS.FirstName, BLOOD\_TYPE.BloodGroup FROM DONOR\_TRANSACTION

INNER JOIN DONORS ON DONOR\_TRANSACTION.DonorID = DONORS.DonorID

INNER JOIN BLOOD\_TYPE ON DONORS.BloodTypeID = BLOOD\_TYPE.BloodTypeID

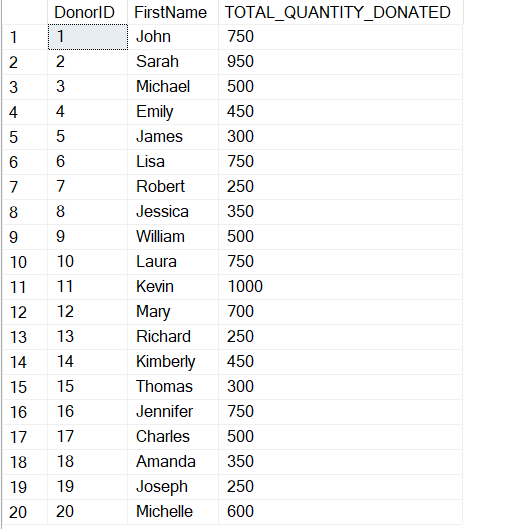


1. Report on total quantity of blood donated by each donor.

SELECT DONORS.DonorID, DONORS.FirstName, SUM(DONOR\_TRANSACTION.DQuantity) AS TOTAL\_QUANTITY\_DONATED FROM DONORS

INNER JOIN DONOR\_TRANSACTION ON DONORS.DonorID = DONOR\_TRANSACTION.DonorID

GROUP BY DONORS.DonorID, DONORS.FirstName

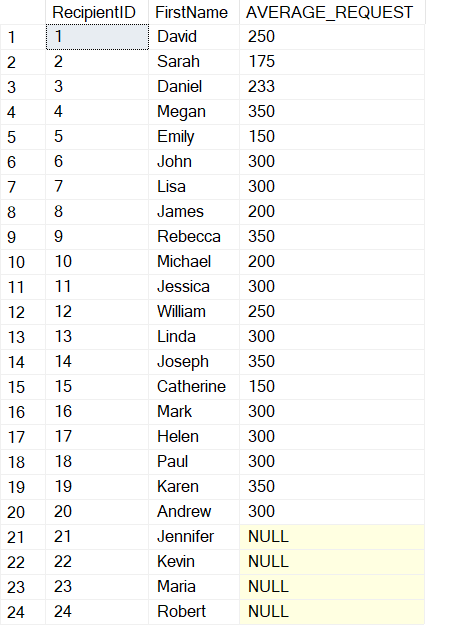
****

1. Report on the average quantity of blood requested by recipients.

SELECT RECIPIENTS.RecipientID, RECIPIENTS.FirstName, AVG(RECIPIENT\_TRANSACTIONS.RQuantity) AS AVERAGE\_REQUEST FROM RECIPIENTS

LEFT JOIN RECIPIENT\_TRANSACTIONS ON RECIPIENTS.RecipientID = RECIPIENT\_TRANSACTIONS.RecipientID

GROUP BY RECIPIENTS.RecipientID, RECIPIENTS.FirstName



1. Report on List donors in alphabetical order by last name.

SELECT FirstName, LastName

FROM DONORS

ORDER BY LastName, FirstName

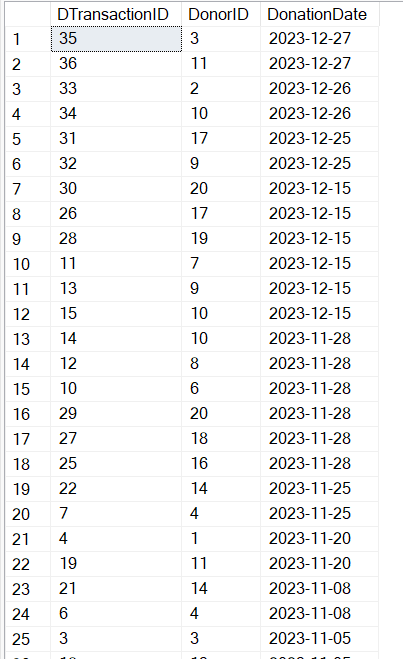


1. Report on donor transactions in descending order of donation date.

SELECT DTransactionID, DonorID, DonationDate

FROM DONOR\_TRANSACTION

ORDER BY DonationDate DESC



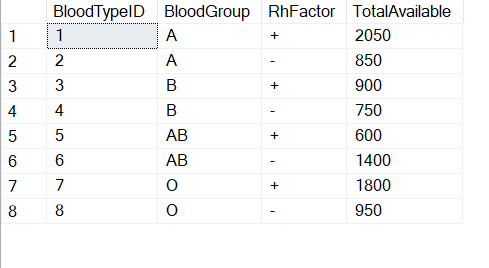
1. Report on the total quantity of blood available for each blood type.

SELECT BLOOD\_TYPE.BloodTypeID, BLOOD\_TYPE.BloodGroup, BLOOD\_TYPE.RhFactor, SUM(BLOOD\_INVENTORY.Quantity) AS TotalAvailable

FROM BLOOD\_TYPE

INNER JOIN BLOOD\_INVENTORY ON BLOOD\_TYPE.BloodTypeID = BLOOD\_INVENTORY.BloodTypeID

GROUP BY BLOOD\_TYPE.BloodTypeID, BLOOD\_TYPE.BloodGroup,BLOOD\_TYPE.RhFactor



1. Report on Donor Information with the Highest Donations

SELECT TOP 10 DONORS.FirstName, DONORS.LastName, DONORS.ContactNumber, BLOOD\_TYPE.BloodGroup, BLOOD\_TYPE.RhFactor,

COUNT(DONOR\_TRANSACTION.DonorID) AS TotalDonations

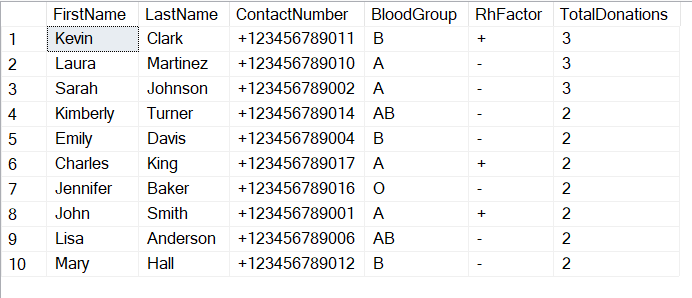
FROM DONORS

INNER JOIN DONOR\_TRANSACTION ON DONORS.DonorID = DONOR\_TRANSACTION.DonorID

INNER JOIN BLOOD\_TYPE ON DONORS.BloodTypeID = BLOOD\_TYPE.BloodTypeID

GROUP BY DONORS.FirstName, DONORS.LastName, DONORS.ContactNumber, BLOOD\_TYPE.BloodGroup, BLOOD\_TYPE.RhFactor

ORDER BY TotalDonations DESC



1. Report showing the workload distribution among staff members at different donation centers.

SELECT DONATION\_CENTRE.CenterName, STAFF.FirstName, STAFF.LastName, STAFF.Position,

COUNT(DONOR\_TRANSACTION.DonorID) AS DonorTransactionsHandled, COUNT(RECIPIENT\_TRANSACTIONS.RecipientID) AS RecipientRequestsHandled

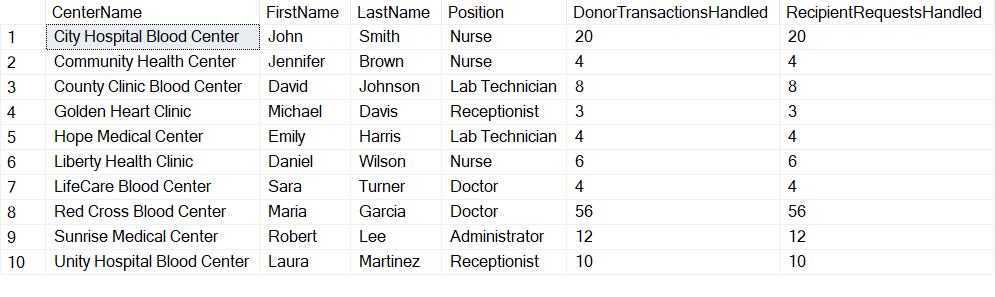
FROM DONATION\_CENTRE

INNER JOIN Staff ON DONATION\_CENTRE.DCenterID = STAFF.DCenterID

LEFT JOIN DONOR\_TRANSACTION ON STAFF.DCenterID = DONOR\_TRANSACTION.DCenterID

LEFT JOIN RECIPIENT\_TRANSACTIONS ON STAFF.DCenterID = RECIPIENT\_TRANSACTIONS.DCenterID

GROUP BY DONATION\_CENTRE.CenterName, STAFF.FirstName, STAFF.LastName, STAFF.Position;



1. Report on list of recipient requests that are still pending and have not been fulfilled.

SELECT RECIPIENTS.RecipientID, RECIPIENTS.FirstName, RECIPIENTS.LastName, BLOOD\_REQUEST.RequestDate, BLOOD\_REQUEST.RequestStatus,

DONATION\_CENTRE.CenterName, BLOOD\_TYPE.BloodGroup, BLOOD\_TYPE.RhFactor

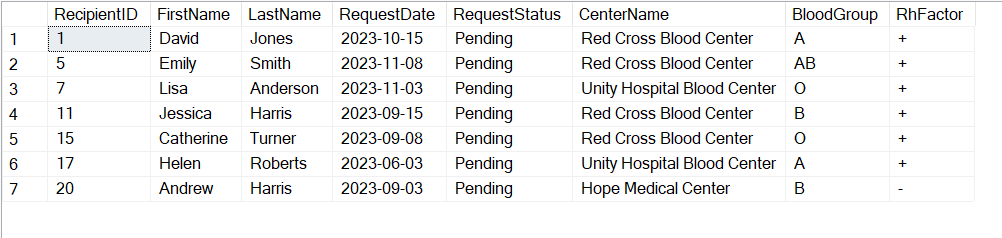
FROM RECIPIENTS

INNER JOIN BLOOD\_REQUEST ON RECIPIENTS.RecipientID = BLOOD\_REQUEST.RecipientID

INNER JOIN DONATION\_CENTRE ON BLOOD\_REQUEST.DCenterID = DONATION\_CENTRE.DCenterID

INNER JOIN BLOOD\_TYPE ON BLOOD\_REQUEST.RequestedBloodType = BLOOD\_TYPE.BloodTypeID

WHERE BLOOD\_REQUEST.RequestStatus = 'Pending';



1. Report on list of recipient requests for a specific blood type and Rh factor.

SELECT RECIPIENTS.FirstName, RECIPIENTS.LastName, BLOOD\_TYPE.BloodGroup, BLOOD\_TYPE.RhFactor,

BLOOD\_REQUEST.RequestDate, BLOOD\_REQUEST.RequestStatus

FROM RECIPIENTS

INNER JOIN BLOOD\_REQUEST ON RECIPIENTS.RecipientID = BLOOD\_REQUEST.RecipientID

INNER JOIN BLOOD\_TYPE ON BLOOD\_REQUEST.RequestedBloodType = BLOOD\_TYPE.BloodTypeID

WHERE BLOOD\_TYPE.BloodGroup = 'A' AND BLOOD\_TYPE.RhFactor = '+';

