Level: 2nd Year Bachelor's in Computer Science (2025/2026)

Module: Databases

TP 4: Relational Algebra

Connect your **Blood-Donation** database to the RAT tool and execute the following queries using Relational **Algebra** based on the following schema:

Donor (**Donor_ID**, First_Name, Last_Name, Gender, Date_Of_Birth, Blood_Type, Phone)

Donation (**Donation_ID**, Donation_Date, Volume_ML, Donation_Type, ***#Donor_ID***)

Blood_Inventory **(Unit_ID**, Blood_Type, Collection_Date, Expiration_Date, Status, ***#Donation_ID***)

Recipient(**Recipient_ID**, First_Name, Last_Name, Gender, Date_Of_Birth, Blood_Type, Hospital_name, Town)

Transfusion (**Transfusion_ID**, Transfusion_Date, ***#Recipient_ID, #Unit_ID***)

1- List the Unit_ID, Blood_Type, and Expiration_Date of all blood inventory units that will expire before February 1, 2025."

$\Pi_{\{ \text{Unit\_ID, Blood\_Type, Expiration\_Date} \}}(\sigma_{\{ \text{Expiration\_Date} < '2025\text{-}02\text{-}01' \}}(\text{Blood\_Inventory}))$

2- Which blood units are available and have an expiration date on or before March 7, 2026?

$\sigma\{\text{Status}='\text{Available}' \wedge \text{Expiration\_Date} \leq '2026\text{-}03\text{-}10'\}(\text{Blood\_Inventory})$

3- Now, change the status of the obtained results from 'Available' to 'Expired' using SQL.

```
UPDATE Blood_Inventory
SET Status = 'Expired'
WHERE Status = 'Available'
AND Expiration_Date <= '2026-03-10';
```

4- Which blood units were collected in 2025?

$\sigma\{'2025\text{-}01\text{-}01' \leq \text{Collection\_Date} \wedge \text{Collection\_Date} \leq '2025\text{-}12\text{-}31'\}(\text{Blood\_Inventory})$

5- How can we record in SQL new blood donations for Donors 1 and 2 on March 10, 2026, and then update the blood inventory with the corresponding blood units, including their blood types and expiration dates (1 month)?"

```
INSERT INTO Donation (Donation_ID, Donation_Date, Volume_ML, Donation_Type, Donor_ID)
VALUES
(129, '2026-03-10', 500, 'Whole Blood', 1);

INSERT INTO Donation (Donation_ID, Donation_Date, Volume_ML, Donation_Type, Donor_ID)
VALUES
(130, '2026-03-10', 500, 'Whole Blood', 2);
```

Then we update the Blood_Inventory table:

```
INSERT INTO Blood_Inventory (Unit_ID, Blood_Type, Collection_Date, Expiration_Date, Status,
Donation_ID)
VALUES
(1029, 'O+', '2026-03-10', '2026-04-10', 'Available', 129);


INSERT INTO Blood_Inventory (Unit_ID, Blood_Type, Collection_Date, Expiration_Date, Status,
Donation_ID)
VALUES
(1030, 'A+', '2026-03-10', '2026-04-10', 'Available', 130);
```

6- List the first_name, last_name, and blood_type of all donors who have donated plasma.

$\Pi\{first\_name, last\_name, blood\_type\}(\sigma\{donation\_type='plasma'\}(donor \bowtie donation))$

If this query does not work under RAT, you can substitute it with this query (using the Cartesian product):

$\Pi\{first\_name, last\_name, blood\_type\}(\sigma\{donation\_type='plasma' \wedge donor.donor\_id=donation.donor\_id\}(donor \times donation))$

7- Get all available blood units (Unit_Id and Blood_Type) and their corresponding donors

$\Pi\{Unit\_Id, Blood\_Type, Donor.First\_Name, Donor.Last\_Name\}(\sigma\{Status='Available'\}(Blood\_Inventory \bowtie Donation) \bowtie Donor)$

If this query does not work under RAT, you can substitute it with this query (using the Cartesian product):

$\Pi\{Unit\_Id, Blood\_Inventory.Blood\_Type, Donor.First\_Name, Donor.Last\_Name\}(\sigma\{Status='Available' \wedge Blood\_Inventory.Donation\_ID=Donation.Donation\_ID \wedge Donation.Donor\_Id=Donor.Donor\_ID\}(Blood\_Inventory \times Donation \times Donor))$

8- Find all the donors who have donated blood and whose donation has been transfused.

$\Pi\{Donor.Donor\_ID, First\_Name, Last\_Name\}(((Donor \bowtie Donation) \bowtie Blood\_Inventory) \bowtie Transfusion)$

If this query does not work under RAT, you can substitute it with this query (using the Cartesian product):

$\Pi\{Donor.Donor\_Id, First\_name, Last\_Name\}(\sigma\{Donor.Donor\_Id=Donation.donor\_Id \wedge Donation.Donation\_id=Blood\_Inventory.donation\_id \wedge Blood\_Inventory.unit\_id=Transfusion.unit\_id\}(Donor \times Donation \times Blood\_Inventory \times Transfusion))$

9- Provide the blood type, first name, and last name of donors and recipients involved in transfusions where their blood types are compatible but not identical

Π { donation.donation_ID, Donor.first_name, donor.last_name, donor.blood_type,Recipient.First_Name, Recipient.Last_Name, recipient.blood_type}(σ {Recipient.Blood_Type ≠Donor.Blood_Type} ((Donor ⋈ Donation ⋈ Blood_Inventory ⋈ Transfusion ⋈ Recipient))

If this query does not work under RAT, you can substitute it with this query (using the Cartesian product):

Π { donation.donation_ID, Donor.first_name, donor.last_name, donor.blood_type,Recipient.First_Name, Recipient.Last_Name, recipient.blood_type}(σ {Recipient.Blood_Type ≠Donor.Blood_Type ∧ Donor.Donor_ID = Donation.Donor_ID ∧ Donation.Donation_ID = Blood_Inventory.Donation_ID ∧ Blood_Inventory.Unit_ID = Transfusion.Unit_ID ∧ Transfusion.Recipient_ID = Recipient.Recipient_ID} (Donor × Donation × Blood_Inventory × Transfusion × Recipient))

)