**Project Title:** Blood Donation System

**Name:** Mounika Ojili

**GitHub URL:** git@github.com:mouniojili/assignment-5.git

**Date:** 1th May 2022

**INITIAL PROPOSAL:**

I'm the proprietor of the Blood Bank. We as the association are helping the destitute patients by giving them the matched blood bunch benefactors for blood. The database stores the giver's record, patient record, record of donor and record of acceptor of blood. This multitude of records are useful to keep the track in blood donation centre up-to-date. This information assists with working on my business and give patients enough measure of blood in the difficult and time of crisis. It is exceptionally useful to track down the donor of blood in difficult time. The information likewise assists us with keeping doing awesome individuals on schedule. The Blood Bank related information is entered by my own system. The patient records are being gotten from the clinic/hospital or centres where blood is required. Likewise, the acceptance of blood information is gotten by these centres and clinics. The donor and donation information are assembled by the concurred individuals who gave the blood to our blood donation centre. Bank workers utilize this information to speak with the contributors in the hardships. The information is extremely useful in building great correspondence with the contributors so they reaction rapidly when the need shows up. This information is additionally utilized for refreshing the bank director about each side by staying up with the latest.

**Relational Database Design Process:**

For my business, I will have the following entities:

1. Blood Bank Info – describes the blood type, Employee ID#, Quantity on Hand, Address and Description.
2. Blood Info – describes Blood types, Blood Bag number, and Code.
3. Hospital Info – Name, Address, Contact Number.
4. Donor Info – describes donor ID#, name, donation date, address, and Contact number.
5. Patient Info – describes patient ID#, name, acceptance date, address, and Contact number.

|  |  |
| --- | --- |
| **Blood Bank Info** |  |
| Blood Type | Varchar[10] |
| Employee\_ID | Integer |
| Quantity\_onHand | Integer |
| Address | Varchar[255] |
| Description | Varchar[255] |

|  |  |
| --- | --- |
| **Blood Info** |  |
| Blood Type | Varchar[10] |
| Blood\_Bag\_Number | Integer |
| Code | Integer |

|  |  |
| --- | --- |
| **Hospital Info** |  |
| Name | Varchar[100] |
| Address | Varchar[255] |
| Contact\_Number | Varchar[20] |

|  |  |
| --- | --- |
| **Donor Info** |  |
| Donor\_ID | Integer |
| Name | Varchar[100] |
| Donation\_Date | Date |
| Address | Varchar[255] |
| Contact\_Number | Varchar[20] |

|  |  |
| --- | --- |
| **Patient Info** |  |
| Patient\_ID | Integer |
| Name | Varchar[100] |
| Acceptance\_Date | Date |
| Address | Varchar[255] |
| Contact\_Number | Varchar[20] |

**UML Class Diagram:**

**Diagram

Description automatically generated**

**ERD Diagram:**

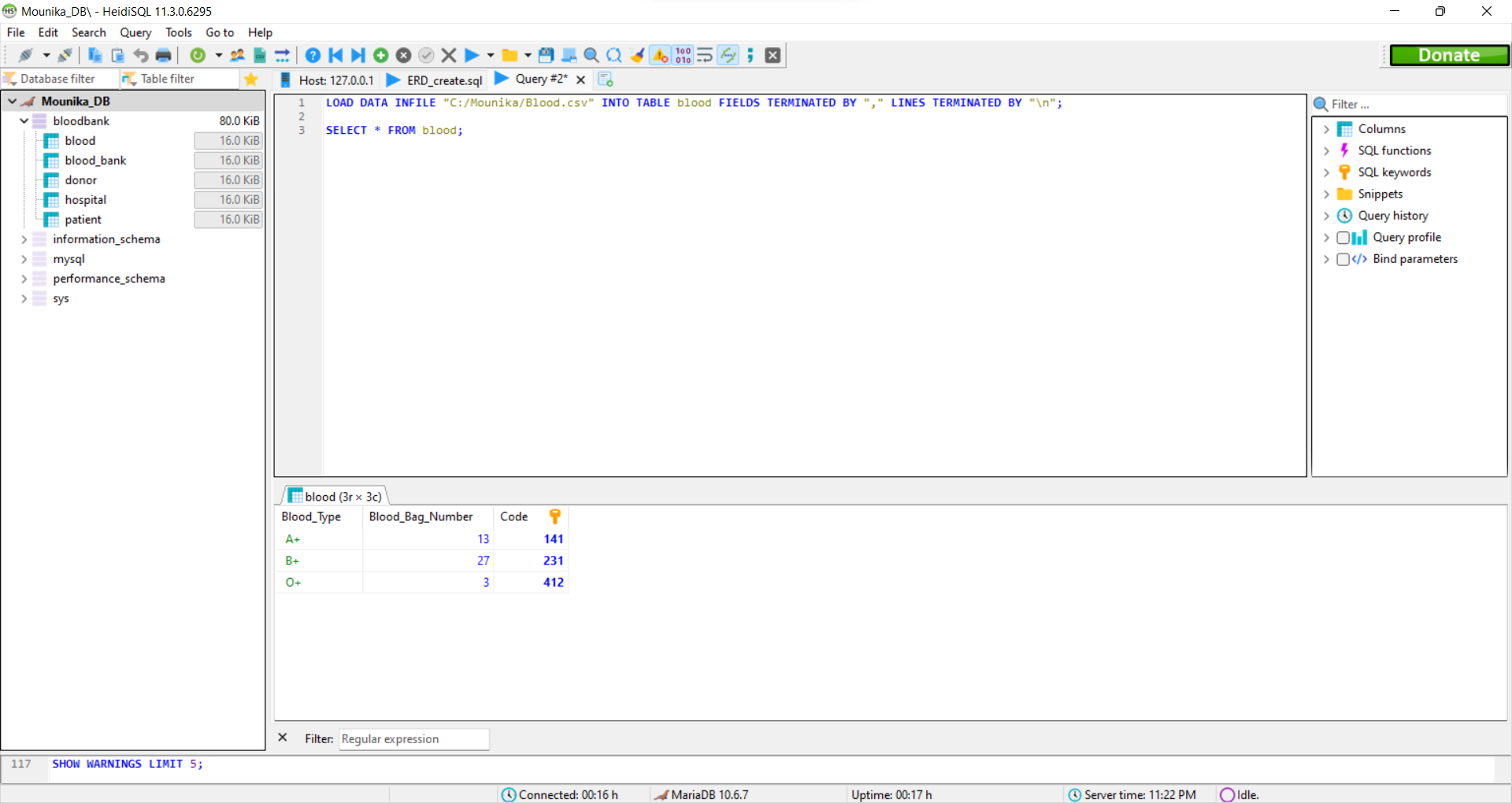
Diagram

Description automatically generated

**Tables Data:**

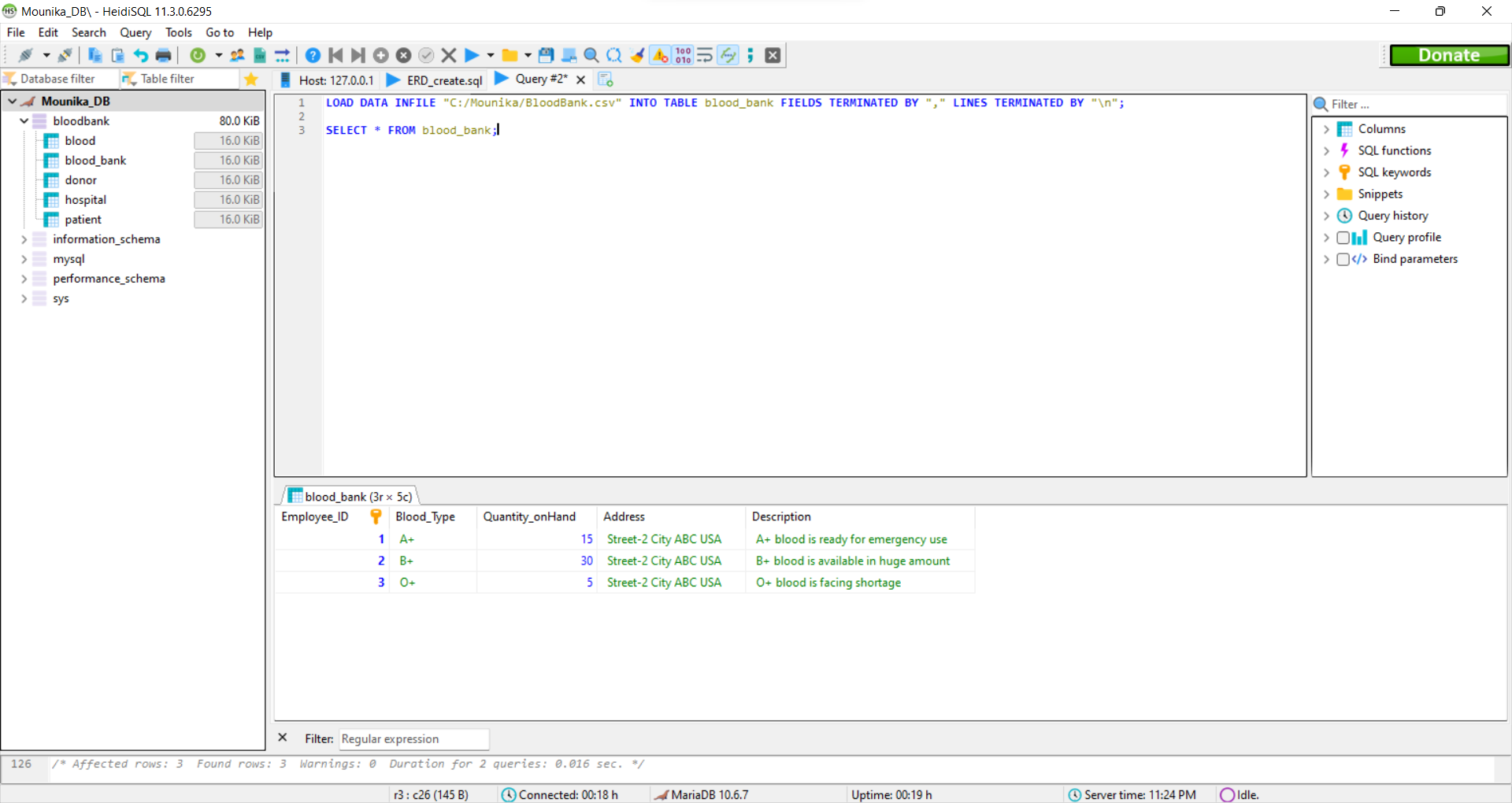
Blood table:

Importing data from Blood.csv file to blood table.



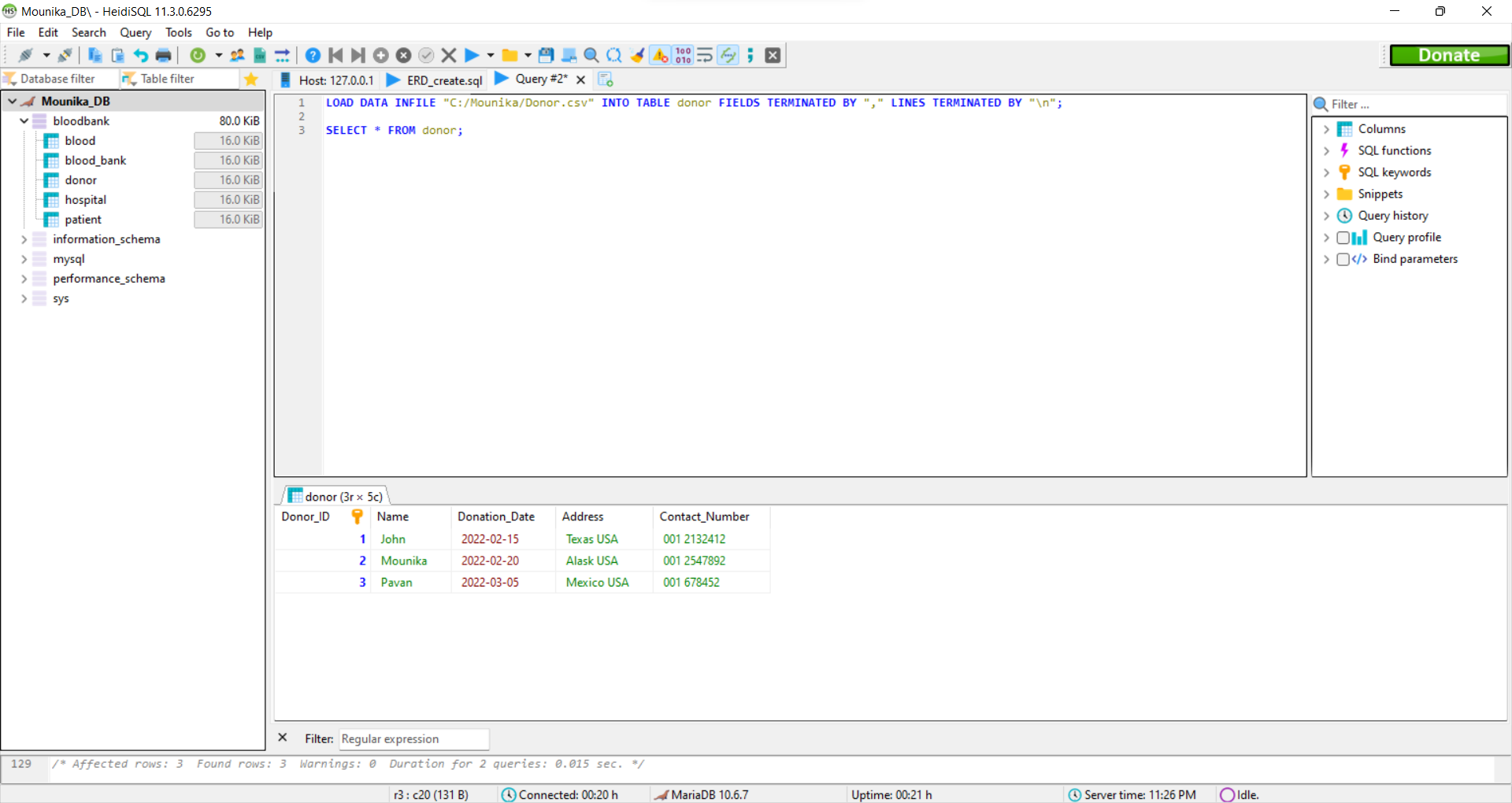
blood\_bank table:

Importing data from Bloodbank.csv file to blood\_bank table.



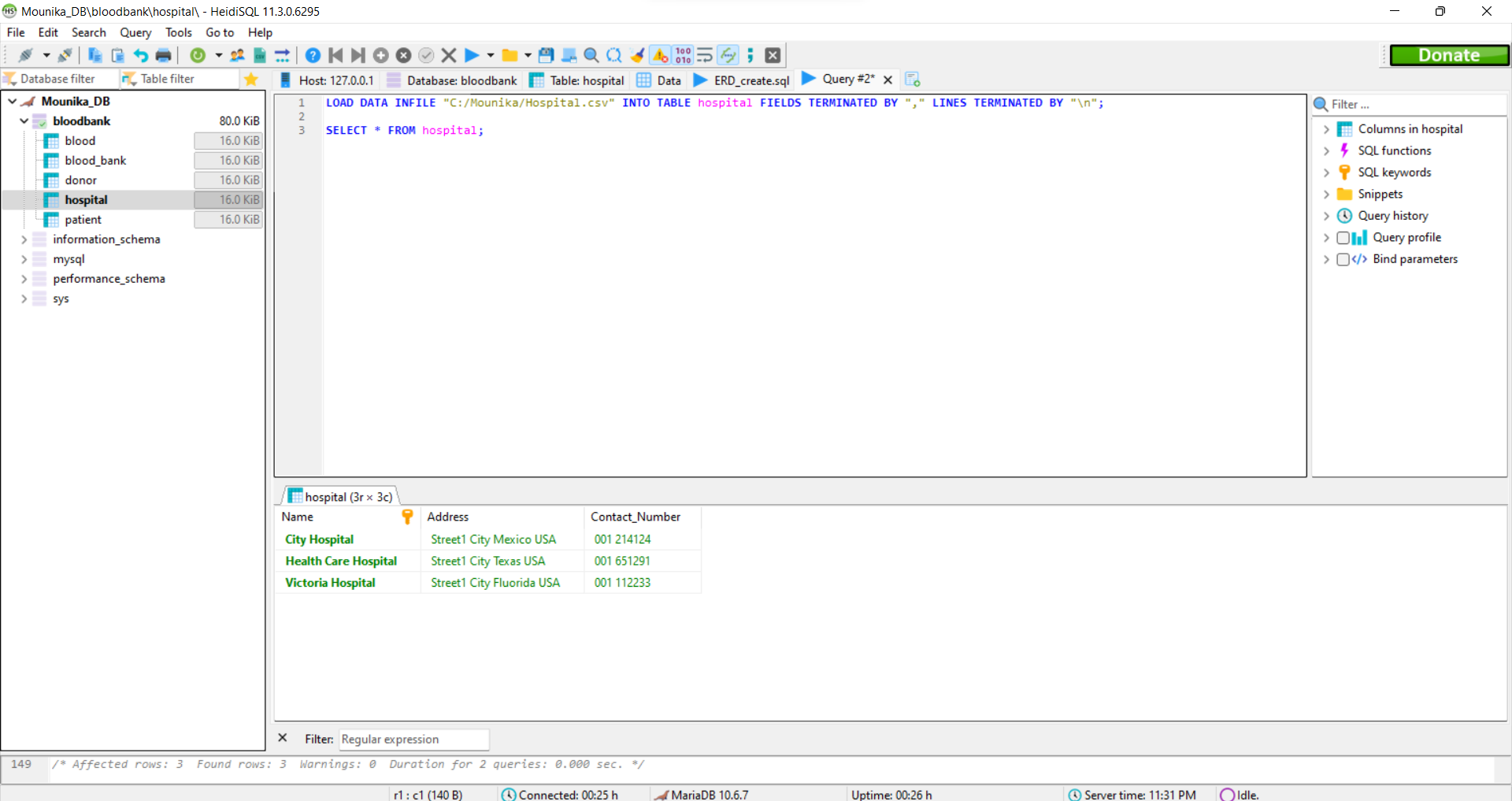
Donor table:

Importing data from Donor.csv file to donor table.



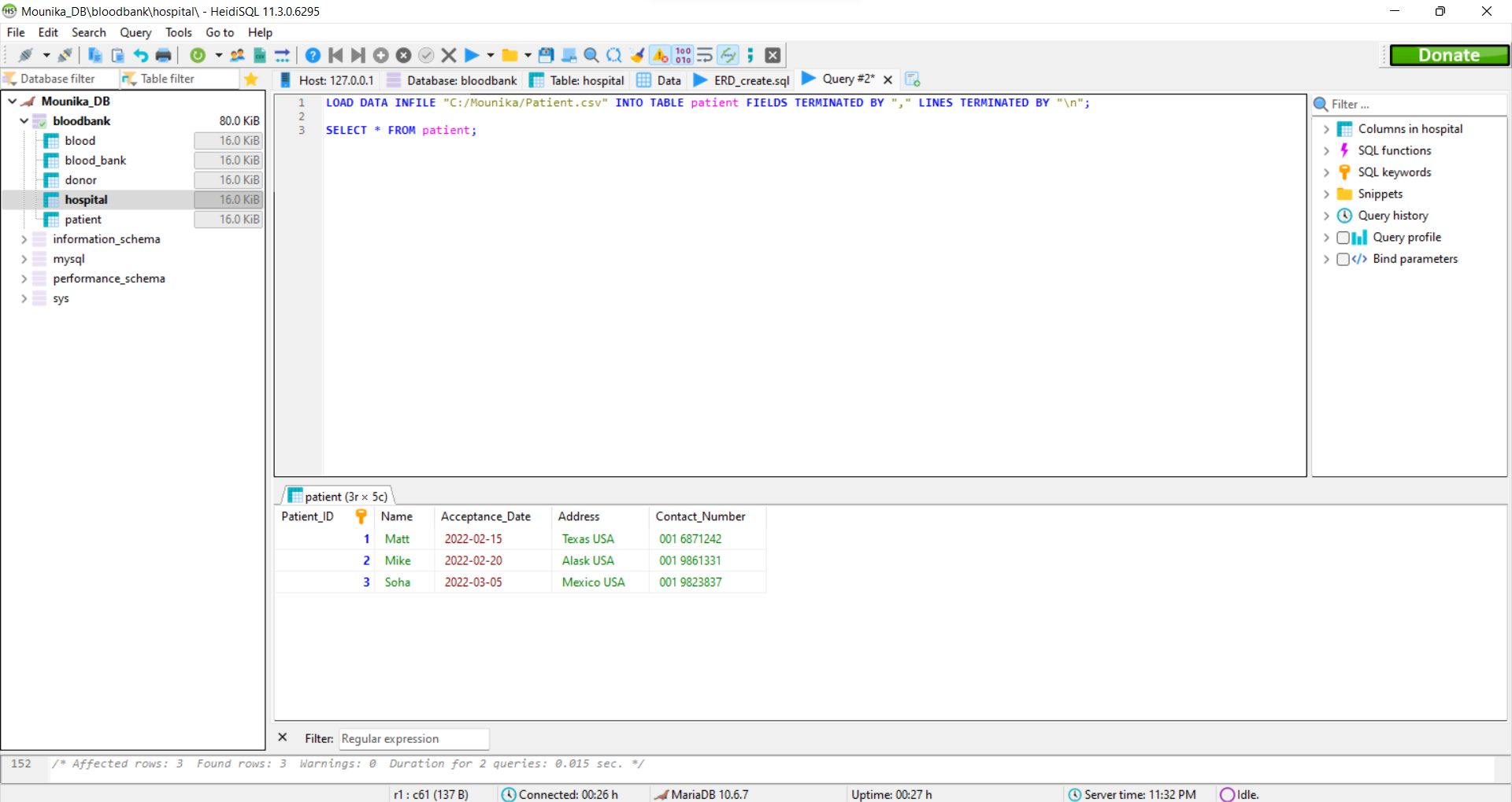
Hospital table:

Importing data from Hospital.csv file to hospital table.



Patient table:

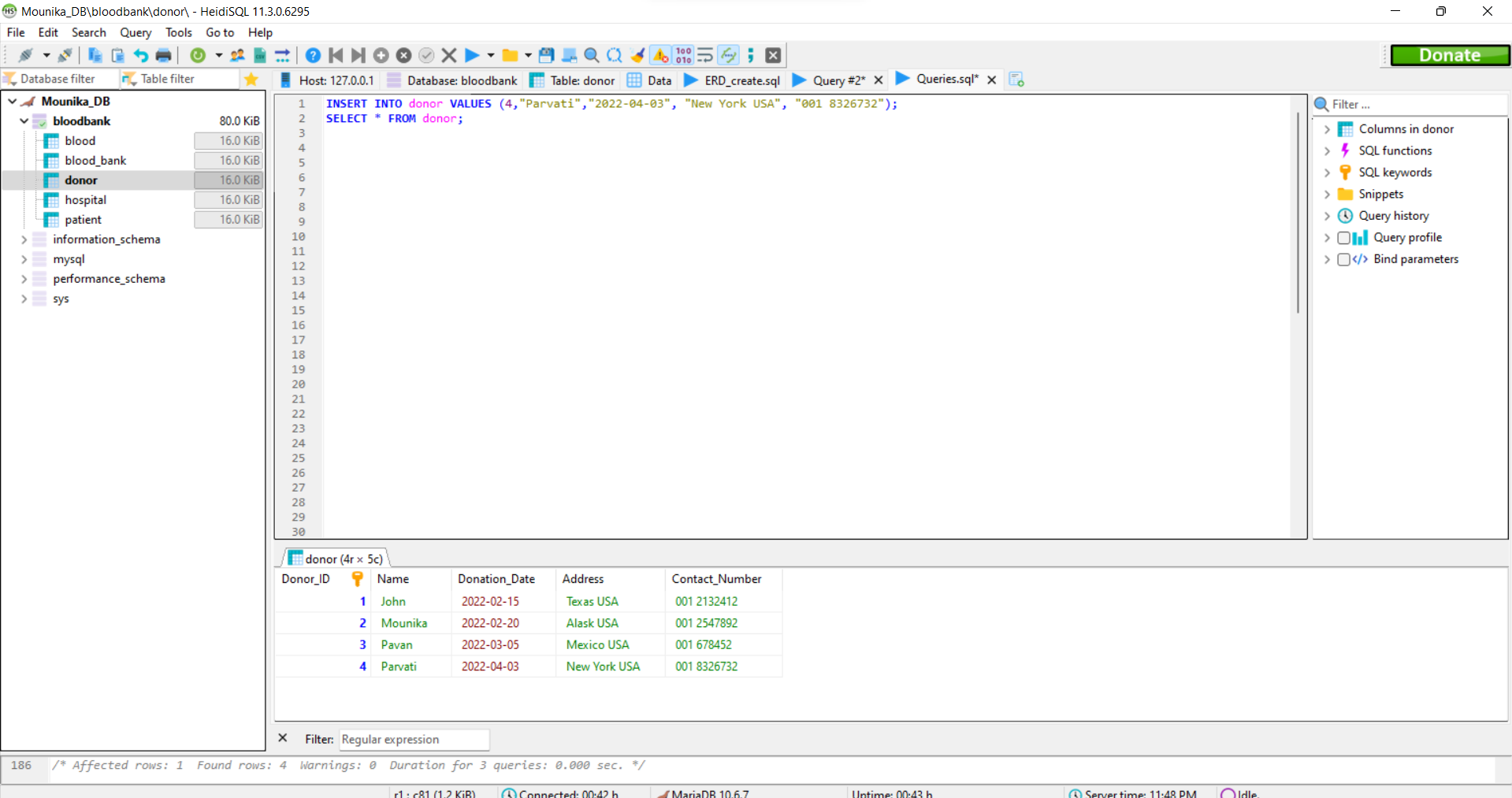
Importing data from Patient.csv file to patient table.



**Table Queries:**

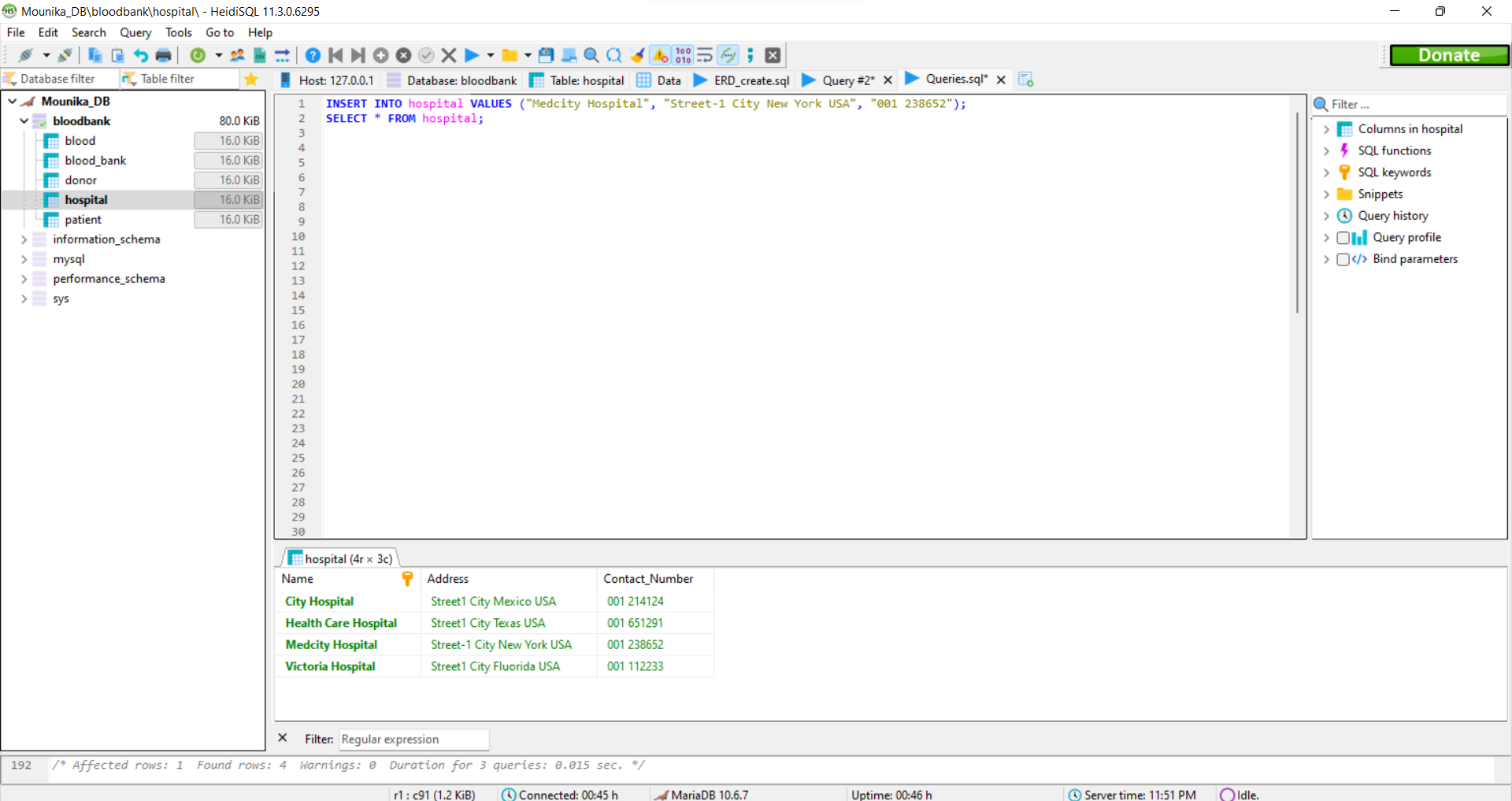
*INSERT INTO donor VALUES (4,"Parvati","2022-04-03", "New York USA", "001 8326732");*

*SELECT \* FROM donor;*



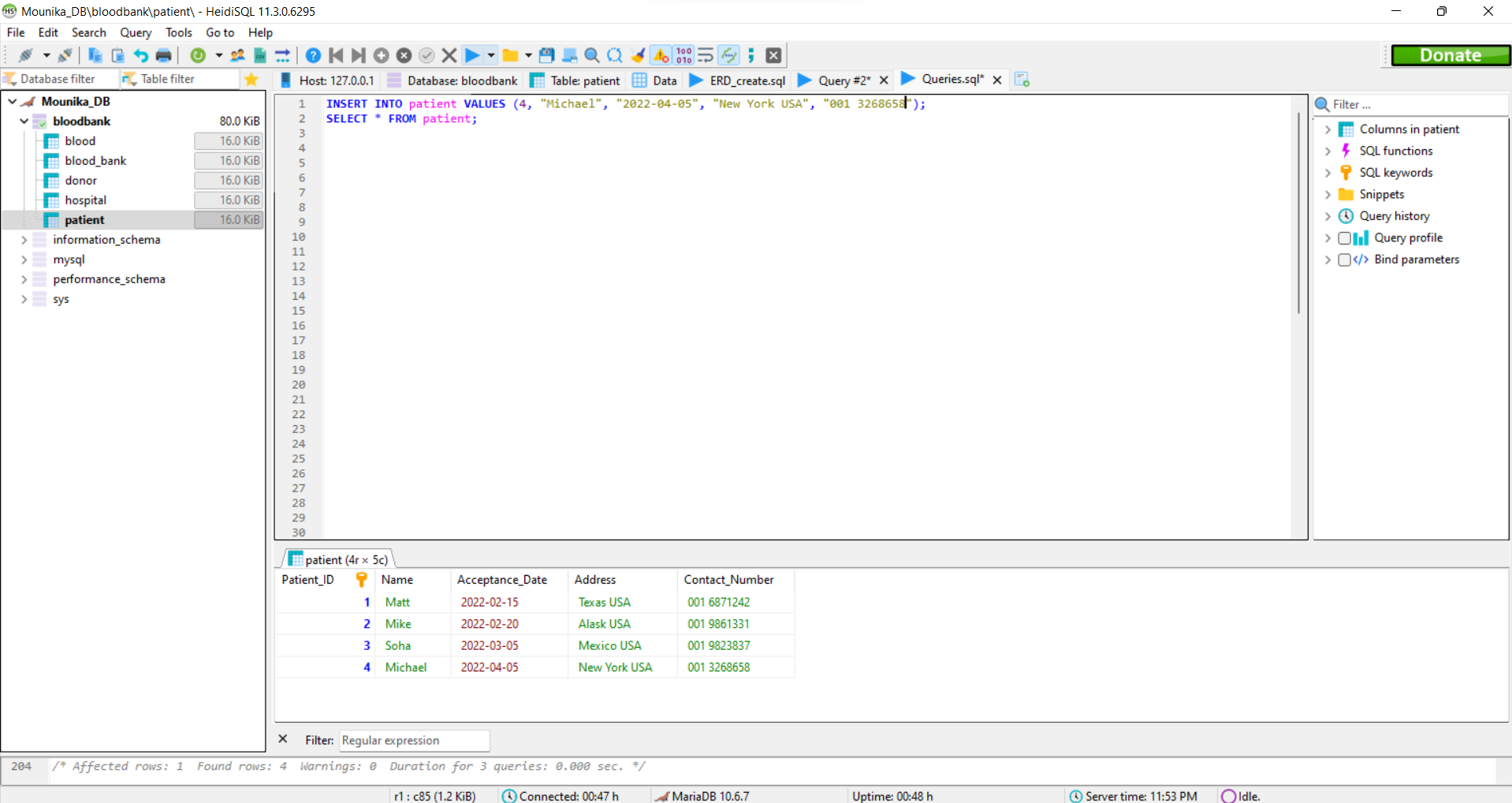
*INSERT INTO hospital VALUES ("Medcity Hospital", "Street-1 City New York USA", "001 238652");*

*SELECT \* FROM hospital;*



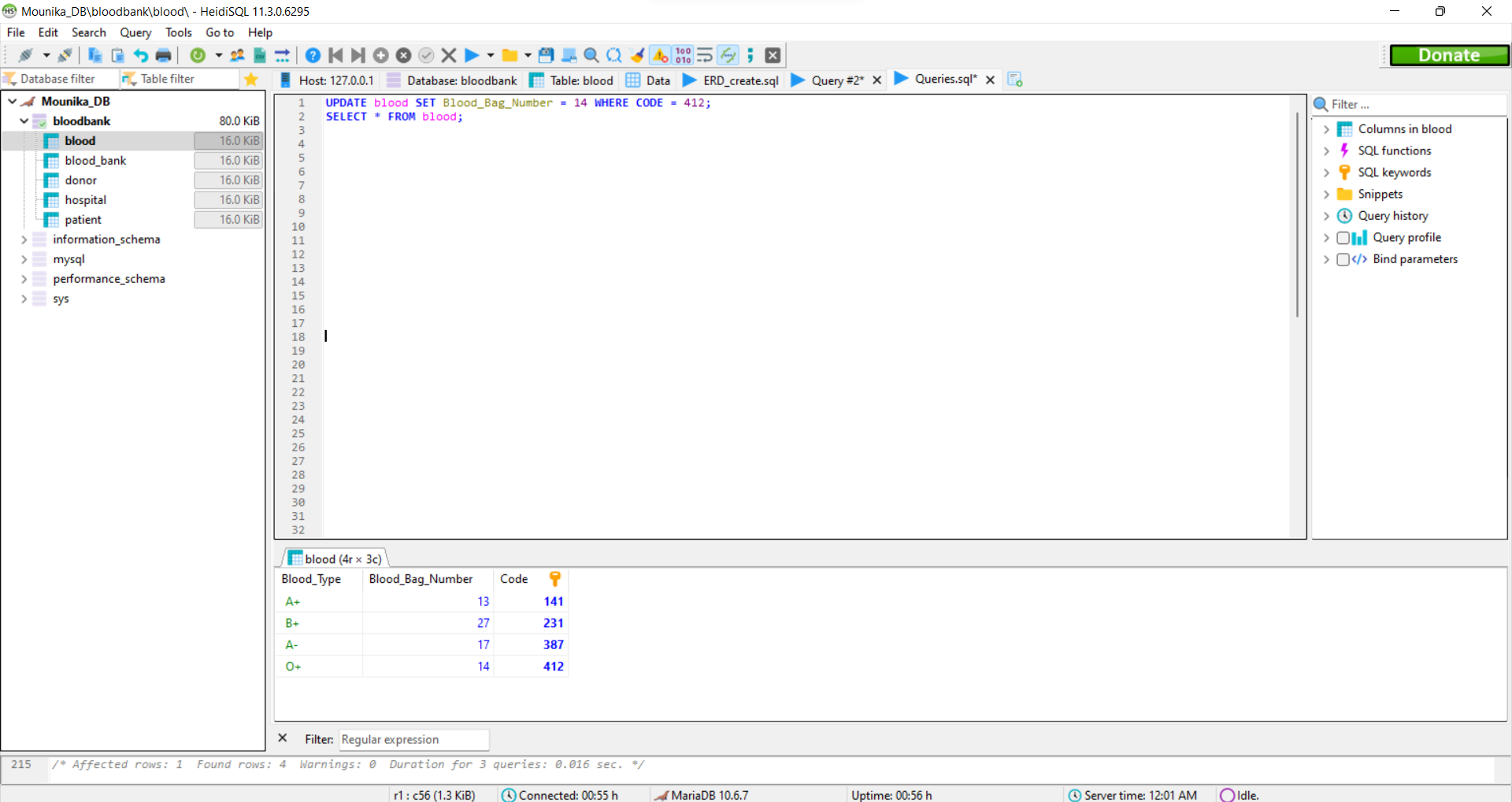
*INSERT INTO patient VALUES (4, "Michael", "2022-04-05", "New York USA", "001 3268658");*

*SELECT \* FROM patient;*



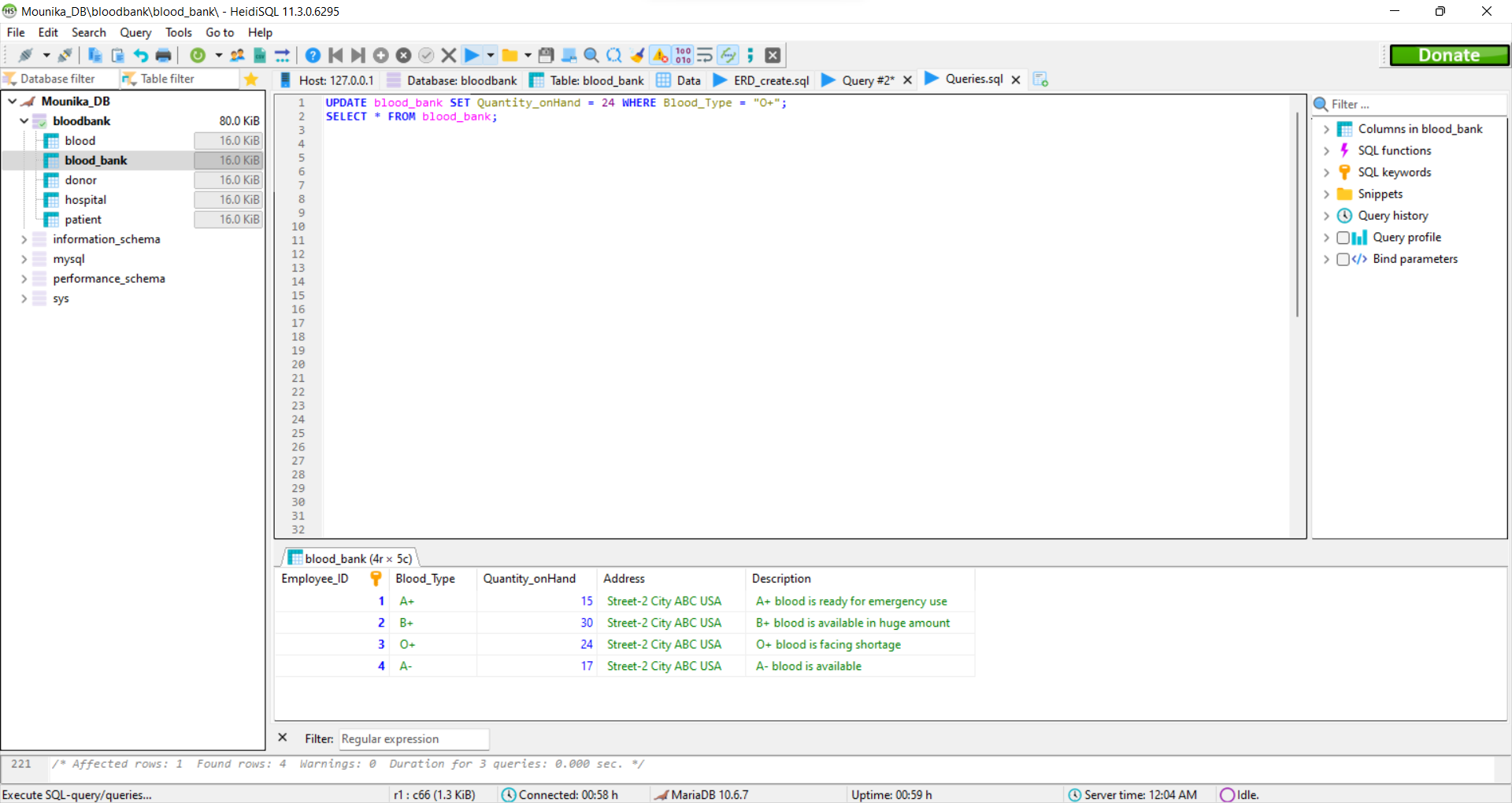
*UPDATE blood SET Blood\_Bag\_Number = 14 WHERE CODE = 412;*

*SELECT \* FROM blood;*



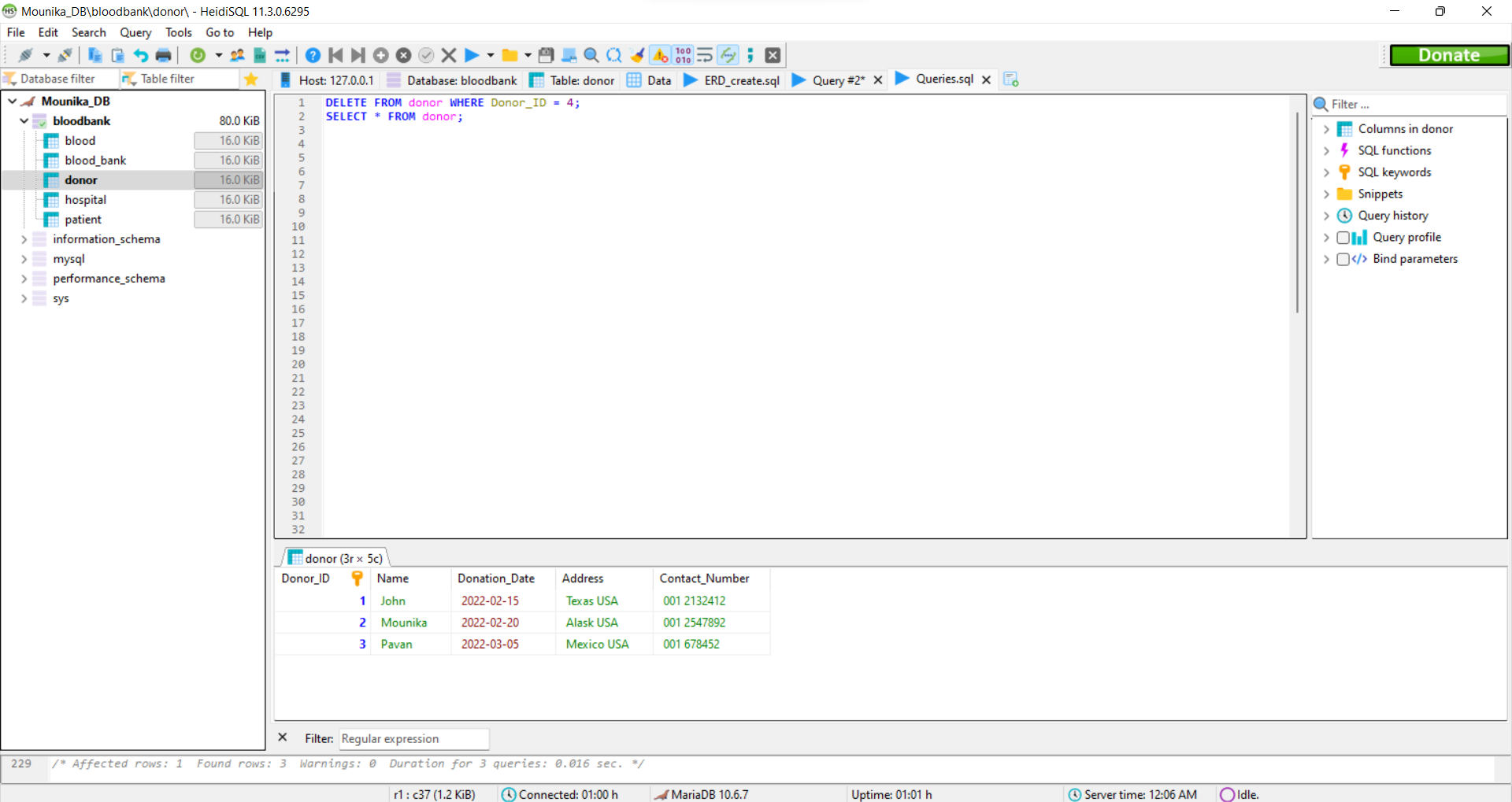
*UPDATE blood\_bank SET Quantity\_onHand = 24 WHERE Blood\_Type = "O+";*

*SELECT \* FROM blood\_bank;*

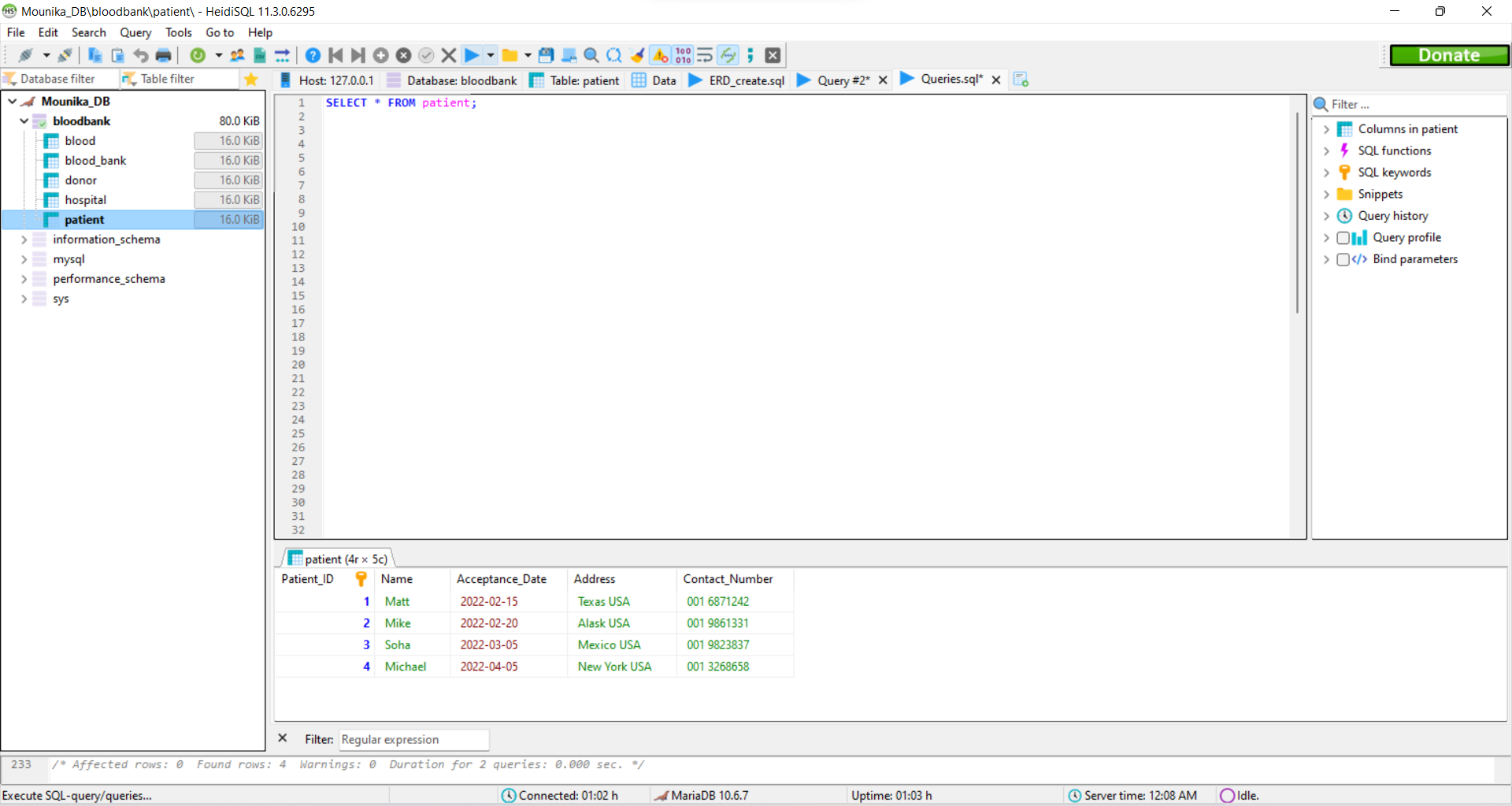


*DELETE FROM donor WHERE Donor\_ID = 4;*

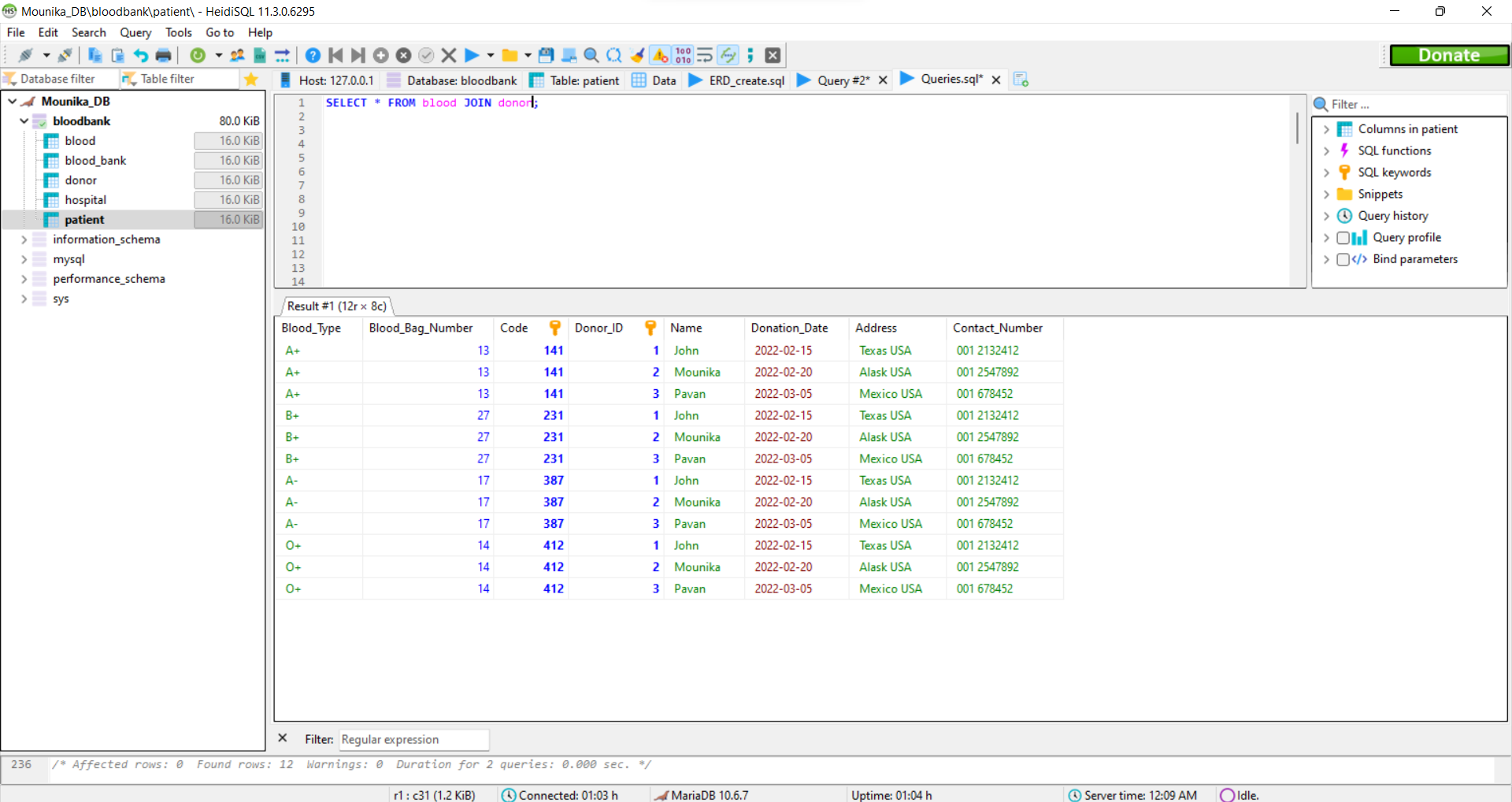
*SELECT \* FROM donor;*



*SELECT \* FROM patient;*



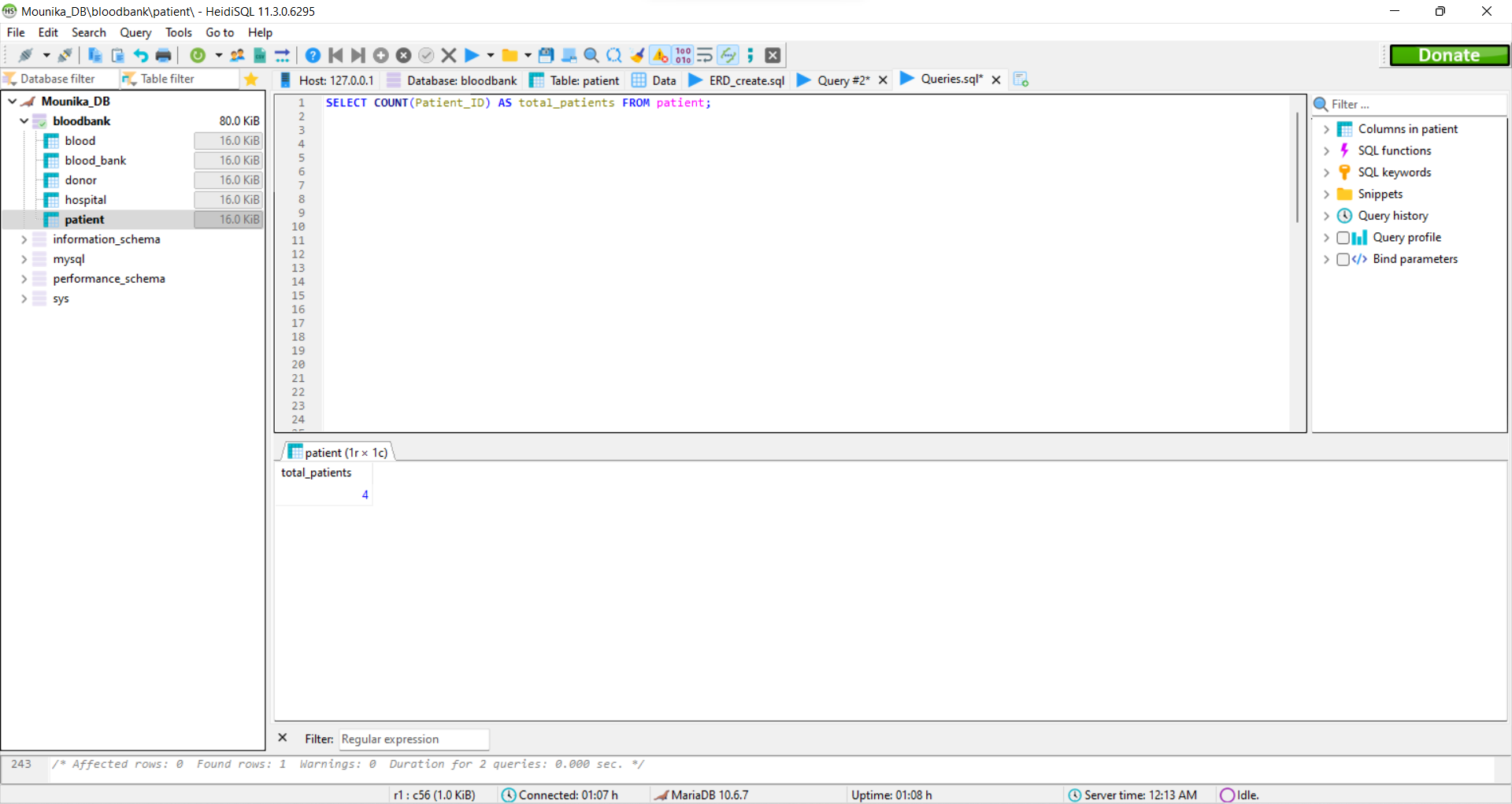
*SELECT \* FROM blood JOIN donor;*



*SELECT \* FROM hospital JOIN patient;*



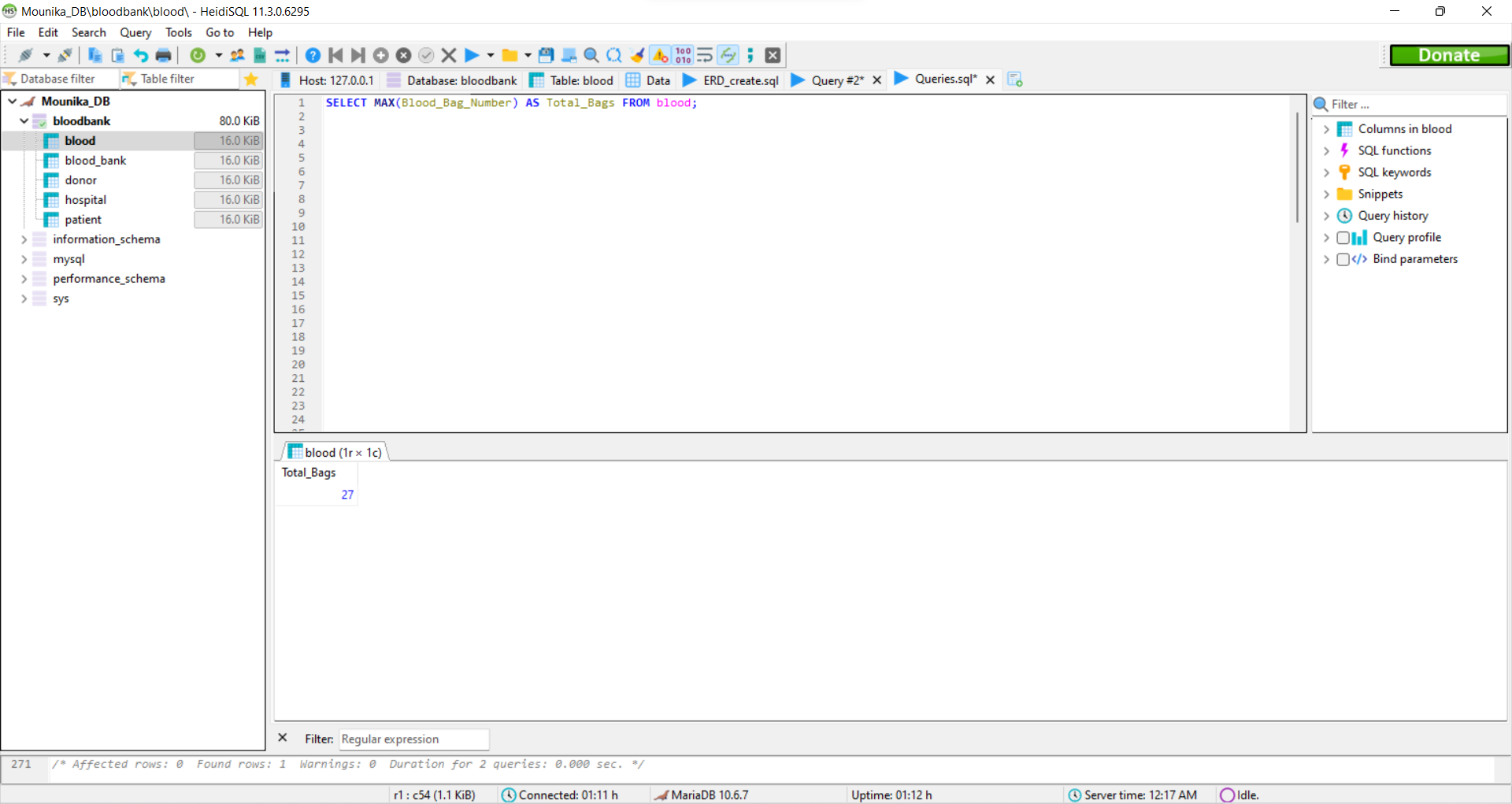
*SELECT COUNT(Patient\_ID) AS total\_patients FROM patient;*



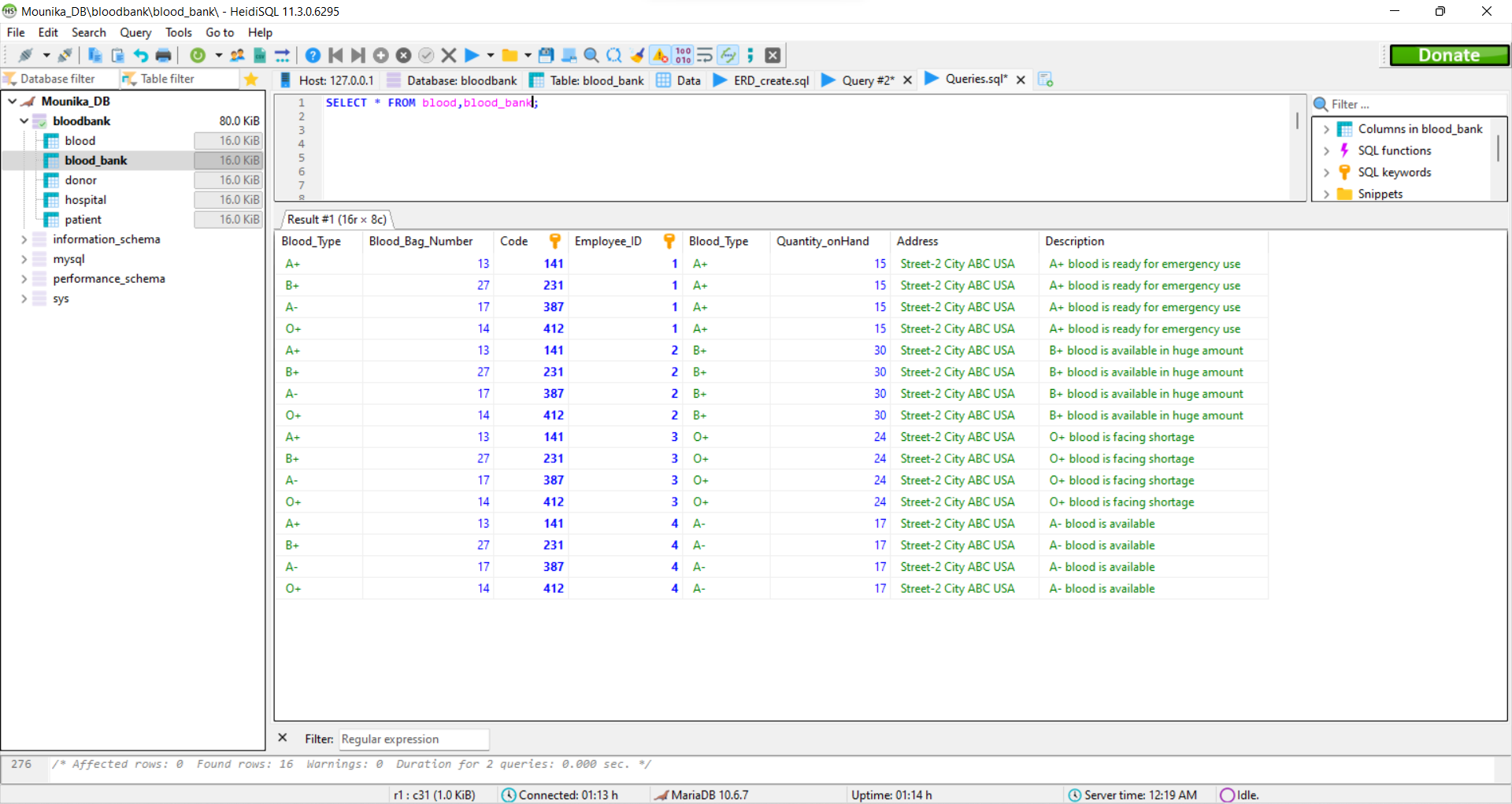
*SELECT SUM(Quantity\_onHand) AS total\_quantity FROM blood\_bank;*



*SELECT MAX(Blood\_Bag\_Number) AS Total\_Bags FROM blood;*



*SELECT \* FROM blood,blood\_bank;*



Index # 1:

I added the index to the patients table because I frequently used this table to check records of blood acceptance by any of the patient. It will boost the results of the query I performed on it.

**CREATE** **INDEX** Date\_Acceptance\_Blood **ON** patient (Acceptance\_Date);

Index # 2:

I added the index to the donors table because I frequently used this table to check records of blood donation by any of the donor. It will boost the results of the query I performed on it.

**CREATE** **INDEX** Date\_Donation\_Blood **ON** donor (Donation\_Date);

These indices are used to speed up data search and SQL query performance. It reduces the number of data pages that must be read to find the specific record. So, I choose them over acceptance record and donation records because these two are the major parts of my database.

View # 1:

**CREATE** **VIEW** donor\_addresses **AS SELECT** **Name**,Address **FROM** donor;

**SELECT** \* **FROM** donor\_addresses;

Graphical user interface, text, application

Description automatically generated

View # 2:

**CREATE** **VIEW** BloodDetails **AS SELECT** Blood\_type,Blood\_bag\_number **FROM** blood;

**SELECT** \* **FROM** BloodDetails;

Graphical user interface, text, application

Description automatically generated

These views can give a us a personalized view of the database structure according to our needs. We created the first view for the blood types and their number of bags we have to keep the total record of stock. Also, we created the view that consists of name and addresses of donors, in case, if we want to approach them in future or contact them. Also, these addresses can help us to send them mail or something else efficiently. We can directly share this kind of view to the service provider so that they could read themselves. We do not even worry about the leakage of our private and confidential data.