IT214 Database Management Systems
Autumn 2020-21

Final Project Report **Section 6 Team 1**

Team ID: Section 6 Team 1

Project Title: Blood Bank Database

Team Members and IDs:

1. Visaj Nirav Shah - 201801016
2. Aayush Desai - 201801060
3. Nishant Shah - 201801403
4. Meet Shah - 201801434

Mentor TA: Mr. Nitish Yadav

Section 1: Final Version of SRS

Software Requirements Specification

for

Blood Bank Management Database

Version 4.0

Prepared by Section 6 Group 1

DA-IICT, Gandhinagar

04/12/2020

Revision History

Name	Date	Reason For Changes	Version
SRS V1	26/09/2020	Initial Document	1.0
SRS V2	01/10/2020	Interviews, Questionnaire, System Requirements, etc. added	2.0
SRS V3	03/10/2020	Suggestions of the supervising TA were incorporated, some additional details inserted	3.0
SRS V4	03/12/2020	Final changes incorporated based on the final database design	4.0

A. Introduction

1. Description of the Problem Statement

1.1 Purpose

The main purpose behind this project is to create a blood bank database that can effectively manage the various services and facilities like donor details, staff information, operating hours, and so on provided by a blood bank. This database should encapsulate all the various features available in a real-life equivalent. This is the Version 4 documentation for this product.

The vision behind creating this product is to increase the functionality of blood banks, which is a critical healthcare facility. Blood banks enter the picture in crucial cases where the patient is in critical condition and a quick response is essential. Maintaining an efficient database can help us give such a response with everything stored and organized in our database.

Quite often, customers are not aware of the availability of blood groups at a particular blood bank, they might not know the rules and protocols to be followed, the schedule of blood camps, and so on. Such information can be quickly updated by the staff and made available instantly to the users.

With this product, we also aim to simplify the tasks of staff and management of the blood bank. They can easily access the required data, efficiently query information, make changes and updates to the system, and so on. Given an easy interface and user-friendly platform, even people not well-versed in database management can use the functionalities without much difficulty.

1.2 Intended Audience and Reading Suggestions

The intended audience for this document is mainly software developers working in the Healthline industry. Other users can include the users of this database like database administrators of the blood bank, staff, owners, managers, and so on. People not directly related to this database like the marketing department, customers themselves, etc. can also refer to this document for a better understanding and retrieval of information from the database.

1.3 Product Scope

This feature-heavy database provides multidimensional functionality. It includes features for various kinds of people including customer features like

availability of blood categorized by different blood groups, blood camp, and its location details, report details, other staff member features like blood camp requirements, inventory, and developer features like easy access to database management, efficient querying, central data access point. Such a product should help us improve the current healthcare management infrastructure facilities, especially in government facilities.

1.4 Description

This blood bank database is a self-contained product i.e. it is an independent system in itself without reliance on any other superset database. The product is aimed at servicing one single blood bank and maintaining their data. What are the features that we expect in a blood bank database?

A central database for all the data storage. The entire data of the blood bank right from the inventory to the staff details can be found in a single database. This one-point access system ensures that there are no data inconsistencies, searching and querying is extremely efficient and easy, and so on. This efficiency is essential for a blood bank database because blood banks usually face emergency situations where a delay of a split-second could put someone's life at risk.

Various components of the blood bank ecosystem are connected to a single system. This means that management and transactions in everyday life can be easily assimilated and performed. For example, inventory can easily be updated based on use and camp allotments. This is also beneficial for administrative work and paper-work procedures since everything can be easily submitted and printed from a single software.

Roles and privileges can be easily assigned since the admin can control the various facets of database management directly. One can easily verify or approve data with documents to approve a user for any given role. With such access restrictions and controls in place, the security of the database is maintained since there is an extremely low probability of errors and false updates to the database.

With such authorization functions, many other related functionalities can be provided. For example, employee details, updating sensitive information, and much more can be done in a systematic fashion only by the authorized personnel. Management of such data, which is required almost daily, is made easy, and can be used without specific knowledge of database management.

Inventory can easily be controlled and updated quickly. The staff needs to simply update the incoming and outgoing blood packets so that the inventory is managed properly. In case a particular product is no longer available, it can quickly be ordered or requested from another source. Since data is stored in such a systematic manner, we can quickly guide our efforts in the required directions.

The database can be used to better understand users. For example, how many donors belong to a particular blood group, how many donated at a particular blood donation camp, etc. The blood bank can direct its marketing and awareness efforts accordingly to bring more people to the donation camp and ease the process of supplying blood. Blood banks can be made more accessible since such data can also be posted online for the use of external visitors.

Blood banks are in constant touch with the hospitals for various reasons. Hospitals ask the patient's family to get blood, and hence they work closely with the blood bank. Hospitals keep track of to whom the blood is donated, under the supervision of a doctor. So, in a way, doctoral supervision over the working of the blood bank increases. They can assess the quality of blood, can check the suitability, blood group status, haemoglobin, sugar level, RBC and WBC, platelets, and so on based on the report. This additional supervision is beneficial for both the hospitals and the blood bank.

An important purpose of this database is to develop the current infrastructure in terms of software and utility for the system to be much more efficient. Usually, blood banks are run by the government or trusts, and because of lack of funds, they might not be technologically sufficient for handling such delicate healthcare tasks. With a central database like this, they can expect a certain level of standard. This database is also crucial since we need to maintain such a huge amount of data. The system should be scalable enough to accommodate the increasing number of database users and functions.

This is a SQL database and has a table-relation schema. The fully-functional database will be of assistance to all the components of the blood bank ecosystem - donors, patients, admin, staff, doctors. Let's analyze the various users and their roles, attributes, and so on to better understand their functions.

The database will be of use at every stage of blood donation to blood transfusion. We have identified the following users and their functions in the database. The primary keys are serial in nature.

Donors are important for any blood bank since they ‘supply’ the blood for future use. Donors would register themselves to the database. Each user will be provided with a unique id and password. Their name, address, and contact number are collected so that they can be contacted in future, if needed. They need to be given various details like blood reports, donation details, scheduling updates about future blood donation camps, etc. Donors can set preferences regarding how their blood is used and to whom it can be provided. The preferences that can be set include personal (can be used only for you), relatives (only for relatives), and general (for anyone).

Patients are elements of the hospital database (not directly in the blood bank database), but they are important for the blood bank to keep track of where their supplies go. Name, address, and contact number of patients are maintained in case they are needed at a later stage. The details of donors are kept anonymous for the patient depending on the donor’s preferences.

Staff will be the everyday user of this product, so many of the features are designed keeping them in mind. The staff members will also have a login from which they can easily access, modify and edit the entries of blood donations and blood reports. They will also be able to manage the inventory of blood pouches, injections, blood testing kit, refreshments, etc. which will help in the management of blood camps. They are divided into teams for specific activities like blood donation camps, other duties so that an efficient number of employees are needed and confusion over duties and responsibilities can be avoided. Various positions are available like manager, nurse, wardboy, janitor, compounder based on the roles and responsibilities assigned.

Admin is the overall controller and in-charge of the blood bank. They have the authorization and privileges power to assign roles (and hence access to specific parts of the system). They are also responsible for scheduling blood donation camps and other events for the blood bank.

Doctors are also a key part of this database, they will be able to check the blood availability with some filters, they can request for the required blood and also insert the data about the patients for whom the blood is used. Doctoral supervision is critical and a necessity for proper functioning of blood donation and transfusion activities since a little error on anyone’s part could be a question of life and death for the patient.

The above descriptions talk about the users of the database. Now, we focus on the entities like blood packet, blood report, inventory, and so on that are a part and parcel of the blood bank.

For every blood packet that is donated, we maintain a record by assigning it a unique ID and noting its other details like from which camp it was taken, who was the donor, details of blood sample, etc. Each blood sample received has blood reports associated with it to maintain a record of blood group, haemoglobin levels, and so on for doctor's and patient's use. This donation is then tagged with the person who donated and blood reports of the sample.

The stocks of various items which are used in everyday functioning of the blood bank and in organizing the blood donation camps are maintained. These are extremely useful for the team to get the details of what was used, what was returned, how many syringes, needles remain, etc. Inventory details of each individual camp are also maintained.

The schedule of upcoming and past blood donation camps is available. Donors can check this schedule to plan their blood donation activities accordingly. Team of doctors, nurses, staff, etc. that is assigned to a particular camp can be added so that there is clarity among those associated. This reduces the administrative workload and eliminates confusion.

For each patient who gets the blood from the blood bank, the details of blood transfusion transaction are maintained. These details are of use for blood bank as well as the hospital for their own records. The supervising doctor's details are also added.

Indirect users of the database like the marketing team, public relations team, and so on can also make use of this database by being passive spectators of the data. Although they are not involved in the database, the admin can choose to share the data with them via a static external method.

Fact finding techniques were used for getting an exhaustive list of requirements which need to be incorporated into the database. These techniques included interviewing prospective users (donor, staff, admin, etc.), sending out a questionnaire survey for getting public opinion, compiling various observations from background readings. These enabled us to determine the relations and data that need to be implemented in our product.

B. Requirements Collection and Fact-Finding

1. Background Readings

- https://www.researchgate.net/publication/339032343_Blood_bank_and_Donor_Management_system
- https://www.sciencedirect.com/science/article/pii/S1877042815036940/pdf?md5=37f01f16137d20d31dcd69d71cc16209&pid=1-s2.0-S1877042815036940-main.pdf
- https://www.webmd.com/a-to-z-guides/complete-blood-count#1
- https://www.hematology.org/education/patients/blood-basics/blood-banking-and-donation

2. Information Gathered from Interviews

Interviews

Interview I

Interviewee: Dr. Heena Desai
Designation: Doctor (MBBS)
Interviewer: Mr. Aayush Desai
Designation: Member, Database Development Team
Date: 27/09/2020
Time: 5 PM
Duration: 30 mins

Purpose of Interview:

- To study the basic requirements needed and problems faced by doctors in the existing system.
- To better understand the role of doctors in the functioning of blood banks.
- Evaluate the current features of such a database and discuss where there is a scope of improvement.

Agenda :

- Understand the required features of the database for doctors.
- Discuss the current modalities of the infrastructure
- Deliberate upon possible improvements and feature-enriching
- Role of doctors

Observations:

1. Login Credentials should be different for the doctors, so they can access it securely.
2. Direct contact details should be available so that doctors can get a quick response.
3. The availability of blood and its details should be easily accessible.
   Blood Details:
   A. Blood Group

B. CBC Measure
    1. RBC
    2. WBC
    3. Hemoglobin
    4. Platelets
4. Doctors can sort and filter the blood details based on their attributes. When Doctors enter the blood group of the patient they should get details of all the blood groups which can be accepted by the patient.

Interview II

Interviewee: Mr. Meet Shah
Designation: Donor
Interviewer: Mr. Visaj Shah
Designation: Member, Database Development Team
Date: 29/09/2020
Time: 5 PM
Duration: 40 mins

Purpose of Interview:

- To study the basic requirements needed and problems faced by donors.
- Understand the target audience i.e. donors.

Agenda :

- Understand the required features of the database for donors.
- Inquire about blood donation patterns and frequency of donation.
- Necessary updates regarding reports, schedules, and so on.

Observations:

1. Login credentials must be provided to each donor so that they can securely access their history of donations.
2. Blood banks should take consent from the donor whether they want their details to remain confidential or not.
3. Donors need updates about future blood donation camps to be held.
4. Blood donors want to see the details of their blood donated (i.e whether his/ her blood is deceased or not, allergy).
5. Donors should be able to have their own blood stored in the blood bank for future emergencies. Especially useful for people with specific requirements/ rare blood groups.
6. Donors should have the option of reserving their specific donation for their relatives. For example, blood donated on day X should be given only to their relatives in the future.

Interview III

Interviewee: Mr. Abhilash Kumar **Designation**: Admin
Interviewer: Mr. Nishant Shah **Designation**: Member, Database Development Team
Date: 28/09/2020 **Time**: 7 PM
Duration: 30 mins

Purpose of Interview:

- To study the basic requirements needed by the Admin.
- Understand the blood bank ecosystem - staff details, infrastructure management, storage protocols, and so on.
- Discuss the current system used and what additional modalities are expected from our product

Agenda :

- Understand the required features of the database for Admin.
- Work on the authorization and privileges aspect of the database.
- General discussions regarding the working of the blood bank.

Observations:

1. Send invitation emails for future blood donation camps to donors after it has been at least three months since they last donated.
2. Provision to assign roles to donors, patients, staff, other teams, and so on.
3. Access over the entire database.
4. The requirement of event scheduling rights.

Interview IV

Interviewee: Mr. Aayush Desai **Designation**: Staff
Interviewer: Mr. Visaj Shah **Designation**: Member, Database Development Team
Date: 28/09/2020 **Time**: 7 PM
Duration: 30 mins

Purpose of Interview:

- To study the basic requirements needed by the Staff members of the blood bank.

Agenda :

- Understand the required features of the database for Staff members.

Observations:

1. They are given access to manage the inventory at the blood bank.
2. Staff members can update the blood details of the donors.

2.1 List of Requirements

- Enhance the security of the database
- Support from doctors in terms of quality monitoring
- Proper inventory maintenance
- Tactical division of privileges among different users of the database
- Proper marketing and public relations for publicity and reach

3. Information Gathered from Questionnaire

Questionnaire:-

1. **Which age group do you belong to** - (Single Choice)
    a. Under 18
    b. Between 18 and 25
    c. Between 25 and 35
    d. Between 35 and 45
    e. Between 45 and 55
    f. Above 55

2. **How often do you donate blood** - (Single Choice; Only for 18+)
    a. I have never donated blood
    b. Between 1-3 times every year
    c. Between 4-6 times every year
    d. More than 6 times every year

3. **Do you feel hesitant to donate blood** - (Single Choice; Only for 18+)
    a. Yes
    b. No

4. **How often have you used the services of a blood bank** - (Single Choice)
    a. Never
    b. 1-3 times
    c. 4-6 times
    d. More than 6 times

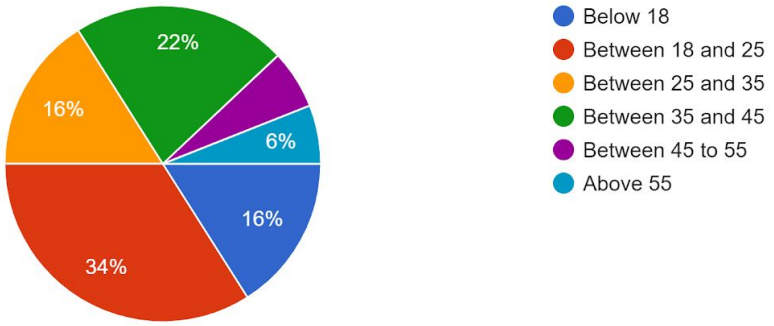
5. **What problems do you face with the current services provided by blood banks** - (Multiple Choice)
    a. Very few in number and not easily accessible
    b. Difficulty in maintaining quality at low costs
    c. Improper inventory management
    d. Government policies are not very supportive of the cause
    e. Administrative procedures and paperwork
    f. Other (User can enter his own explanation)

6. **Suggestions or ideas to better improve the Blood Bank Management System** - (short answer)

3.1 Graphs and Charts Based on Responses Collected

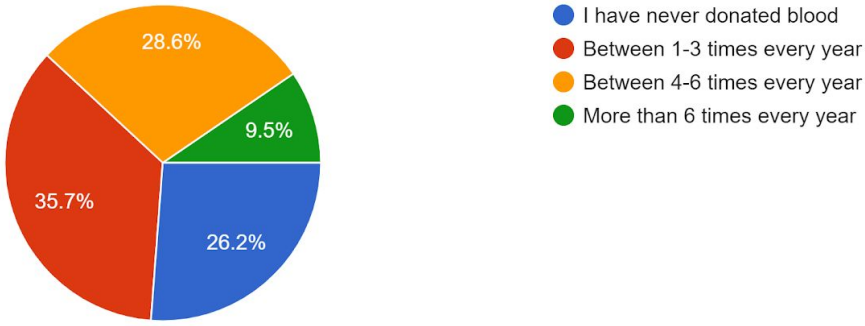
Which age group do you belong to

50 responses



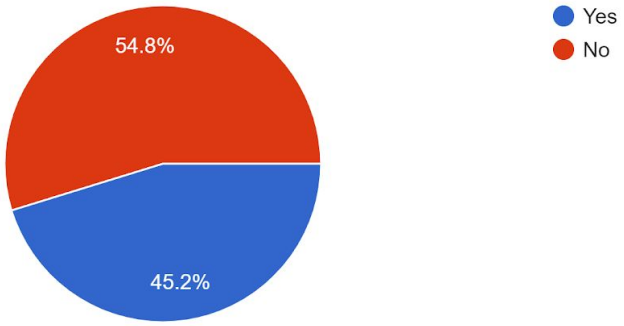
How often do you donate blood?

42 responses



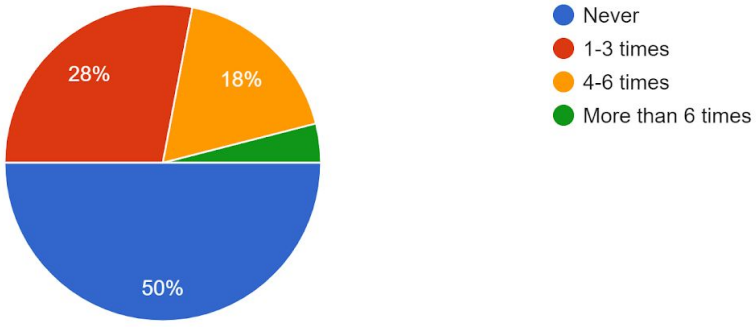
Do you feel hesitant to donate blood?

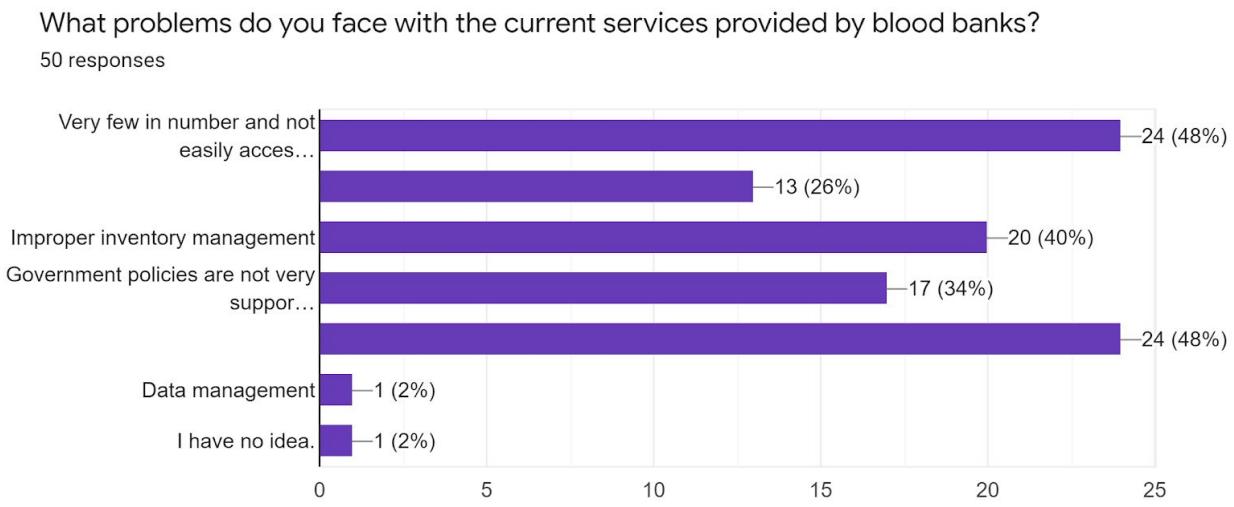
42 responses



How often have you used the services of a blood bank?

50 responses





Short Answers Received

- Inventory details should be kept open to the public. Also, the schedule of blood camps should be available on a public forum.
- I was hospitalized a while back and some patient next to me could not find a bottle of blood because the hospital blood bank did not intimate him in advance. His situation worsened because of this.
- I want to be able to get enough data about how my donated blood is used, who is the beneficiary, etc.
- We need more doctoral supervision for better quality checks.
- I myself organize some blood donation camps and attend many more. I would like to have better infrastructure and equipment so that the collected samples are healthy and clean.
- I used to work at a blood bank and the database there had many faults. The authorization was faulty, some critical features were missing, etc.
- I want to donate blood but I have never been able to.
- We need to spread awareness about the importance of blood donation

3.2 List of Requirements

- Very few in number and not easily accessible
- Difficulty in maintaining quality at low costs
- Improper inventory management
- Government policies are not very supportive of the cause
- Administrative procedures and paperwork
- Details of the blood camps are updated regularly

4. Observations

4.1 Combined Requirements

- Enhance the security of the database
- Tactical division of privileges among different users of the database
- Doctors must have access to check the availability of the blood at the blood bank.
- Blood bank staff must have authority to enter and update the blood report details of the donor.
- Admin has the right to send mails to donors who have last donated 3 months ago.
- Support from doctors in terms of quality monitoring
- Proper inventory maintenance
- Proper marketing and public relations for publicity and reach
- Very few in number and not easily accessible
- Difficulty in maintaining quality at low costs
- Government policies are not very supportive of the cause
- Administrative procedures and paperwork should be reduced
- Details of the blood camps are updated regularly

C. Fact-Finding Chart

Objective	Technique	Subject(s)	Time commitment
Enhance the security of the Interview database	Interview	Mr. Abhilash Kumar	30 mins
Support from doctors in terms of quality monitoring	Interview	Dr. Heena Desai	30 mins
Proper inventory maintenance	Interview	Mr. Meet Shah	40 mins
Tactical division of privileges among different users of the database	Interview	Mr. Abhilash Kumar	30 mins
Proper marketing and public relations for publicity and reach	Interview	Mr. Abhilash Kumar	30 mins
Very few in number and not easily accessible	Questionnaire	Google Form Respondee	1 min
Difficulty in maintaining quality at low costs	Questionnaire	Google Form	1 min

		Respondee	
Government policies are not very supportive of the cause	Questionnaire	Google Form Respondee	1.5 min
Administrative procedures and paperwork	Questionnaire	Google Form Respondee	1 min
Improper inventory management	Questionnaire	Google Form Respondee	2 min
Details of the blood camps are updated regularly	Questionnaire	Google Form Respondee	3 min

D. List Requirements

1. Final List of Requirements with Frequency

- Reduction of paperwork and administrative procedures, especially in emergency cases (Frequency = 30)
- Precise control of inventory with minimal faults in data (Frequency = 25). This is basically maintained by staff and admin.
- Increase the reach of the blood bank - accessibility and awareness (Frequency = 25).
- Unique privileges based on the position and services offered (Frequency = 21)
- Transparency of data - details of the inventory, schedule of blood donation camps, etc. made available easily (Frequency = 23)
- Increased supervision of operations by doctors (Frequency = 3)
- Improve the current infrastructure and make the systems scalable (Frequency = 4)

E. Users

Donors

a. Donors can register for Blood Donation.
b. Donors are asked to enter their medical details.
    - Blood Group (required*)
    - Past Blood Reports
    - Medical History
c. The access to view the past history of their own blood donations.
d. Not allowed to access to view other donor's data.
e. Blood Donation type

- Self
- Family Member
- Open to all

f. Donors are updated regularly about the upcoming Blood Donation Camps.

Admin

a. They are allowed to maintain the entire database. They can make necessary updates and deletions in the database.
b. They have the authority to assign roles like Doctor, Donor, etc. in the databases. They can even revoke the roles under certain circumstances.

Staff

a. They have the right to view the data of all the donors and the recipients.
b. They have access to enter the blood report of the Donor.
c. They can't change the personal details which are entered by the donor or the patient.
d. They are authorized to maintain the request by the doctors.

Doctors

a. They can view the data of blood availability(blood group-wise).
b. They are allowed to request the blood if needed.
c. They don't have any information about the donors.
d. Doctors can enter details of the patient for whom he has requested the blood.

F. Operating Requirements

Hardware Requirements

- Can be accessed from any device which has internet connectivity.

Software Requirements

- We have used SQL.

G. Product Function

- Donors should be allowed to enter personal and medical details.
- Donors can get themselves registered for blood donation.
- Donors are updated about the future blood donation camps.
- Past blood donation details of donors are made accessible of their own.
- Bank your own blood.
- Reserve their own blood for their relatives.
- Staff members can update blood details of donors.
- Staff members can manage the inventory at the blood bank.
- Admin has rights for manual updates if needed.
- Admin has event scheduling rights.
- Admin assigns roles such as Doctors, Staff, Donors.
- It should allow updates when a patient uses blood or a donor donates blood.
- History of all the patients and Donors.
- Doctors will get unique login credentials for security.
- Doctors have contact details of the highest authority of the blood bank for quick response.
- Doctors can view the blood availability data. They have a filter facility.
- Doctors are able to request the blood for the patient.

H. Privileges

Donors

a. They can register themself.
b. View the history of their Blood Donations.
c. Updates on the upcoming Blood Donation camps.

Admin

a. He can intervene for manual updates in the database in critical conditions.
b. He has the right to view the entire database.

Staff

a. Enter the Blood report of the Donor.

b. View the entire data.
c. They maintain requests of blood from the Doctors.

Doctors

a. Request for the Blood.
b. View the Blood availability details.
c. Can request for the blood.

I. Assumptions

- Isolated Model
    - We are assuming no interactions with other blood banks.
- Individual donations, not made in the blood camp, are not accepted.
- A part of a blood packet, not necessarily the whole, can be given to the patient.
- There is no expiry limit for a blood packet donated.

J. Business Constraints

- There are no business constraints.

Section 2: Final Noun Analysis

TABLE 1: Nouns and Verbs in Description

NOUN	VERB
PERSON	DONATION
CREDENTIALS	ORDER
DONOR	SUPPLY
BLOOD PACKET	REQUEST
BLOOD REPORT	TRANSFUSION
PATIENTS	
DOCTOR	
STAFF	
BLOOD DONATION CAMP	
INVENTORY	
STOCKS (BLOOD BANK INVENTORY)	
TEAM	
ITEM	
EMPLOYEE	
BOTTLES	
BLOOD GROUP	
EXTERNAL VISITORS	
HOSPITAL	

GOVERNMENT	
FUND	
USERS	
PASSWORD	
INJECTIONS	
BLOOD TESTING KIT	
REFRESHMENTS	
ROLES	
NAME	
PHONE NUMBER	
BLOOD BANK	
CENTRAL DATABASE	
STAFF DETAILS	
ACCESS SYSTEM	
COMPONENTS	
BLOOD BANK ECOSYSTEM	
PRIVILEGES	
FACETS	
DOCUMENTS	
DOCTOR DETAILS	
PATIENT DETAILS	
SQL DATABASE	
DONATION DETAILS	

POUCHES	
ADMIN	
ADMIN DETAILS	
USERNAME	
USAGE TYPE	
BLOOD TAG	
PERSON	
DESIGNATION	

TABLE 2: Entities, Relations, and Attributes in Database

Candidate Entity Set	**Candidate Attribute Set**	**Candidate Relationship Set**
ADMIN	• ADMIN_ID • NAME • ADDRESS • MOBILE • USERNAME • PASSWORD	
DONOR	• DONOR_ID • NAME • ADDRESS • MOBILE • USERNAME • PASSWORD	• DONOR donates BLOOD PACKET
BLOOD PACKET	• PACKET_ID • CAPACITY	• CAMP_DETAILS (with BLOOD DONATION CAMP)

	• CAMP_ID • DONOR_ID • USAGE_TYPE	• REPORT_OF (with BLOOD REPORT) • DETAILS_OF_PACKET (with TRANSFUSION) • DONOR donates BLOOD PACKET
BLOOD REPORT	• REPORT_ID • PACKET_ID • BLOOD_GROUP • RBC_COUNT • WBC_COUNT • PLATELETS • SUGAR_LEVEL • HAEMOGLOBIN • TIME_STAMP	• REPORT_OF (with BLOOD PACKET) • REPORT_DATA (with DONATION)
PATIENT	• PATIENT_ID • NAME • ADDRESS • MOBILE	
DOCTOR	• DOCTOR_ID • NAME • ADDRESS • MOBILE • USERNAME • PASSWORD	• DETAILS_OF_DOCTOR (with TRANSFUSION)
STAFF	• STAFF_ID • SALARY • POSITION • DATE_OF_JOINING • NAME • ADDRESS • MOBILE • USERNAME • PASSWORD	• WITH_STAFF(WITH TEAM)
BLOOD	• CAMP_ID	• CAMP_DETAILS (with

DONATION CAMP	• EVENT_DATE • LOCATION • START_TIME • END_TIME	BLOOD PACKET) • WITH_TEAM (with TEAM) • FOR_CAMP (with INVENTORY)
INVENTORY	• CAMP_ID • ITEM_ID • COUNT	• FOR_CAMP (with BLOOD DONATION CAMP) • ITEM_NAME (with ITEM)
TEAM	• CAMP_ID • STAFF_ID	• WITH_TEAM (with BLOOD DONATION CAMP) • WITH_STAFF(WITH STAFF)
ITEM	• ITEM_ID • NAME • TYPE • ITEM_COUNT • PRICE	• ITEM_NAME (with INVENTORY)
TRANSFUSION	• PACKET_ID • PATIENT_ID • DOCTOR_ID • TIME_STAMP	• DETAILS_OF_PATIENT (with PATIENT) • DETAILS_OF_DOCTOR (with DOCTOR) • DETAILS_OF_PACKET (with BLOOD PACKET)

TABLE 3: Rejected Nouns with Reason

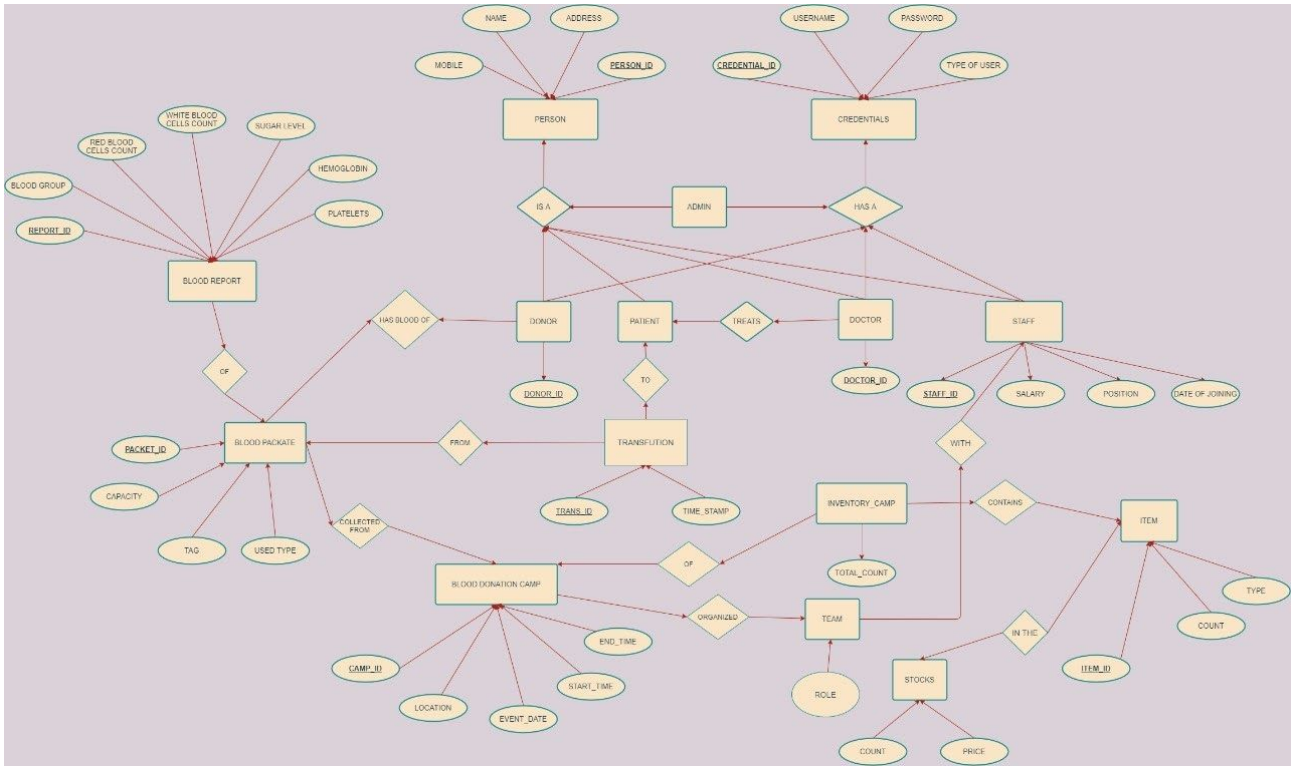
REJECTED NOUN	REJECTION REASON
PERSON	Each role like donor, patient, etc. is used separately
EMPLOYEE	Employee and staff have same role
BOTTLES	Bottles, blood pouches, packets are all similar

EXTERNAL VISITORS	No active use in the database
HOSPITAL	Not directly related
GOVERNMENT	Not required
FUND	Funds are not considered
USERS	Similar to person
INJECTIONS	Will come under ITEM
BLOOD TESTING KIT	Will come under ITEM
REFRESHMENTS	Will come under ITEM
ROLES	Equivalent to DESIGNATION in CREDENTIALS
BLOOD BANK	Database for a single blood bank
CENTRAL DATABASE	No other external database is used or needed
STAFF DETAILS	Covered under STAFF
ACCESS SYSTEM	Not related
COMPONENTS	Equivalent to DESIGNATION in CREDENTIALS
BLOOD BANK ECOSYSTEM	Not related
PRIVILEGES	Equivalent to DESIGNATION in CREDENTIALS
FACETS	Not related
DOCUMENTS	Similar to BLOOD REPORT
DOCTOR DETAILS	Equivalent to DOCTOR
PATIENT DETAILS	Equivalent to PATIENT
SQL DATABASE	No other external database is used

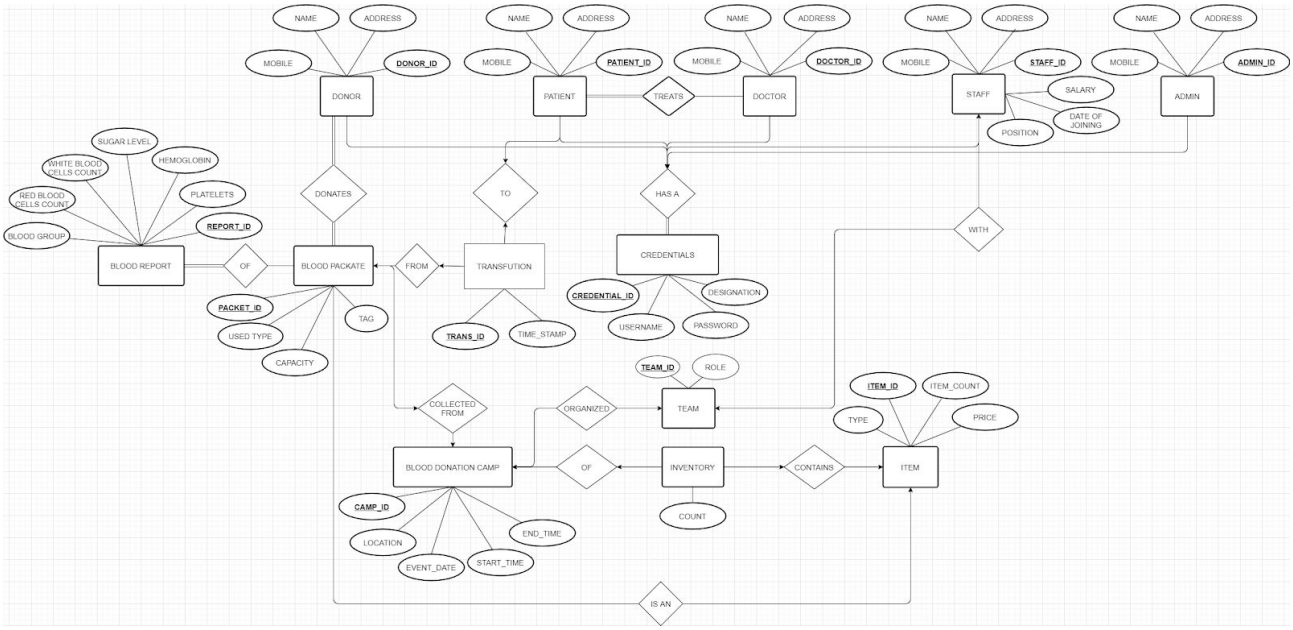
	or needed
DONATION DETAILS	Equivalent to DONATION
POUCHES	Bottles, blood pouches, packets are all similar
CREDENTIALS	Allotted individually to each user

Section 3: Final ER Diagrams - All Versions

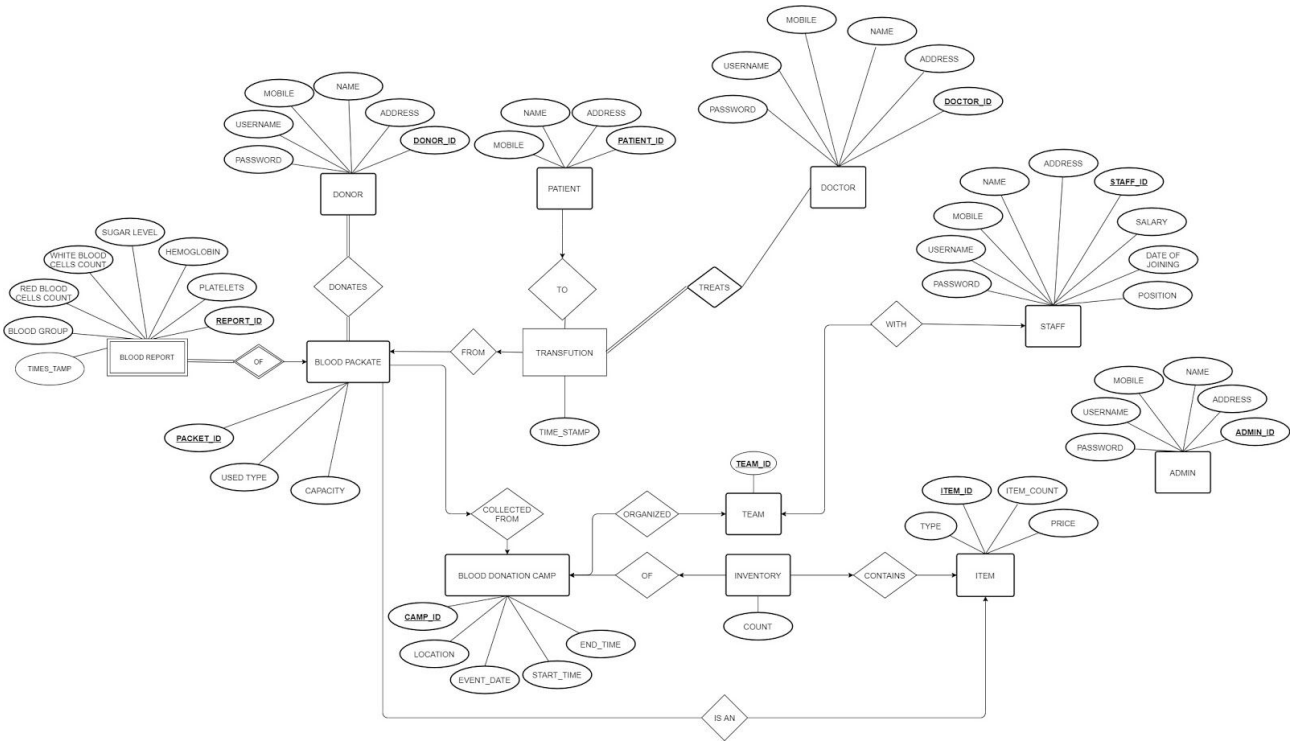
Version 1:



Version 2:



Version 3 - Final:



Section 4: Conversion of Final ER Diagram to Relational Model

Relational Model (before schema refinement and normalization):

Admin(ADMIN_ID,Username,Password,NAME, ADDRESS, MOBILE)
Donor (Donor_ID, Username, Password, Name, Address, Mobile)
Doctor (Doctor_ID,Username,Password,Name,Address,Mobile)
Staff
(Staff_ID,Username,Password,Salary,Credentials_ID,Position,Date_of_joining,Name,Address,Mobile)
Patient (Patient_ID,Doctor_ID,Name,Address,Mobile)
Item(Item_ID,Name, Type, Item_Count)
Inventory (Camp_ID,Item_ID,Count)
Donation_Camp (Camp_ID,Event_date,Location,Start_time,End_time)
Team (Team_ID,Staff_ID, CAMP_ID)
Blood_Packet (Packet_ID,Capacity, Camp_ID, Donor_ID, Usage_Type,Item_ID)
Blood_Report (Report_ID,Packet_ID,Platelets,Hemoglobin,Sugar_level,WBC,RBC,Blood_group)
Transfusion(Trans_ID, Packet_ID,Patient_ID, Doctor_ID, Time_Stamp)

Section 5: Normalization and Schema Refinement

Functional Dependency and Anomalies in original schema

1) ADMIN_ID -> NAME,ADDRESS,USERNAME, PASSWORD, MOBILE
    - BCNF

2) Donor_ID -> Username, Password, Name, Address, Mobile
    - BCNF

3) Doctor_ID -> Username,Password,Name,Address,Mobile
    - BCNF

4) Staff_ID -> Username,Password,Salary,Position,Date_of_joining,Name,Address,Mobile
    - BCNF

5) (Patient_ID) -> Name,Address,Mobile
    - BCNF

6) Item_ID -> Name, Type, Item_Count, price
    - BCNF

7) (Camp_ID,Item_ID) -> Count
    - BCNF

8) Camp_ID -> Event_date,Location,Start_time,End_time
    - BCNF

9) (Staff_ID,CAMP_ID)
    - BCNF

10) Packet_ID -> Capacity, Camp_ID, Donor_ID, Usage_Type
    - BCNF

11) Report_ID ->
Packet_ID,Platelets,Hemoglobin,Sugar_level,WBC,RBC,Blood_group,TIME_STAMP
    - BCNF

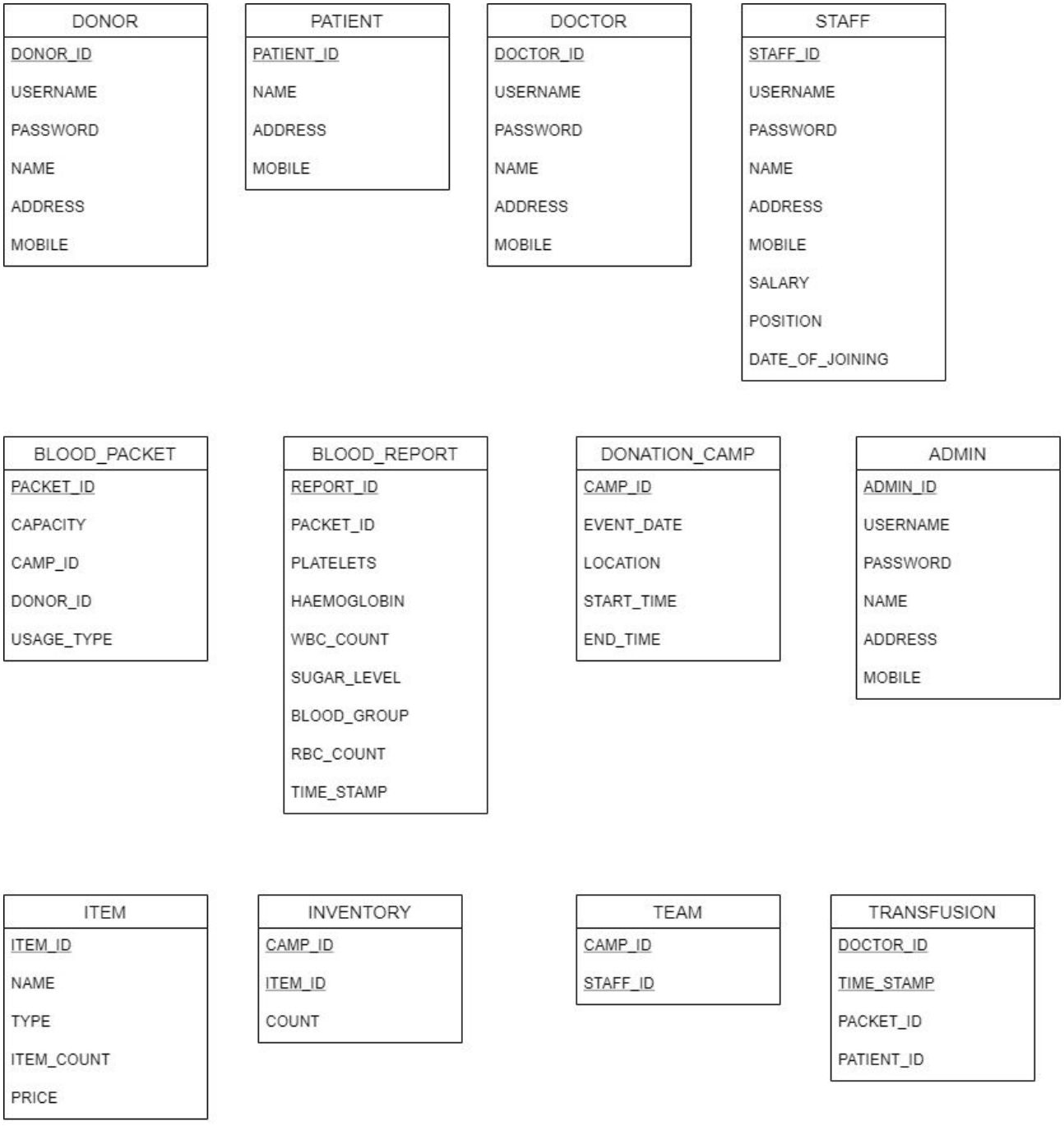
12) Trans_ID -> Patient_ID, Doctor_ID, TIMESTAMP
(Patient_ID, Doctor_ID) -> TIMESTAMP

    - This is not in 3rd normal form because Trans_ID defines (Patient_ID, Doctor_ID) ans (Patient_ID, Doctor_ID) defines TIMESTAMP.
    - For this we REMOVE TRANS_ID and make DOCTOR_ID and TIME_STAMP as primary keys..
    - After this it comes in BCNF

FINAL SCHEMA

Admin(ADMIN_ID,Username,Password,NAME, ADDRESS, MOBILE)
Donor (Donor_ID, Username, Password, Name, Address, Mobile)
Doctor (Doctor_ID,Username,Password,Name,Address,Mobile)
Staff
(Staff_ID,Username,Password,Salary,Credentials_ID,Position,Date_of_joining,Name,Address,Mobile)
Patient(Patient_ID, Name, Address, Mobile)
Item(Item_ID,Name, Type,Item_Count)
Inventory (Camp_ID,Item_ID,Count)
Donation_Camp (Camp_ID, Event_date, Location, Start_time, End_time)
Team (Staff_ID, CAMP_ID)
Blood_Packet (Packet_ID,Capacity, Camp_ID, Donor_ID, Usage_Type, Item_ID)
Blood_Report
(Report_ID,Packet_ID,Platelets,Hemoglobin,Sugar_level,WBC,RBC,Blood_group,TIME_STAMP)
Transfusion(Doctor_ID,TIME_STAMP,Packet_ID,Patient_ID)

Schema Diagram



Section 6: SQL - Final DDL Scripts, Insertion Queries, 40 SQL Queries and Their Snapshots

DDL Script:

```
CREATE TABLE ADMIN (
        ADMIN_ID SERIAL PRIMARY KEY,
        USERNAME VARCHAR(20),
        PASSWORD VARCHAR(20),
        NAME VARCHAR(20),
        ADDRESS VARCHAR(100),
        MOBILE VARCHAR(10)
);

CREATE TABLE DONOR (
        DONOR_ID SERIAL PRIMARY KEY,
        USERNAME VARCHAR(20),
        PASSWORD VARCHAR(20),
        NAME VARCHAR(20),
        ADDRESS VARCHAR(100),
        MOBILE VARCHAR(10)
);

CREATE TABLE DOCTOR (
        DOCTOR_ID SERIAL PRIMARY KEY,
        NAME VARCHAR(20),
        ADDRESS VARCHAR(100),
        MOBILE VARCHAR(10),
        USERNAME VARCHAR(20),
        PASSWORD VARCHAR(20)
);

CREATE TABLE STAFF (
        STAFF_ID SERIAL PRIMARY KEY,
        SALARY INT,
        USERNAME VARCHAR(20),
        PASSWORD VARCHAR(20),
        POSITION VARCHAR(20),
        DATE_OF_JOINING DATE,
        NAME VARCHAR(20),
        ADDRESS VARCHAR(100),
        MOBILE VARCHAR(10)
);
```

```
CREATE TABLE PATIENT(
	PATIENT_ID SERIAL PRIMARY KEY,
	NAME VARCHAR(20),
	ADDRESS VARCHAR(100),
	MOBILE VARCHAR(10)
);

CREATE TABLE ITEM (
	ITEM_ID SERIAL PRIMARY KEY,
	NAME VARCHAR(20),
	TYPE VARCHAR(20),
	ITEM_COUNT INT,
	PRICE INT
);

CREATE TABLE DONATION_CAMP  (
	CAMP_ID SERIAL PRIMARY KEY,
	EVENT_DATE DATE,
	LOCATION VARCHAR(100),
	START_TIME TIME,
	END_TIME TIME
);

CREATE TABLE INVENTORY (
	CAMP_ID SERIAL,
	ITEM_ID SERIAL,
	COUNT INT,
	PRIMARY KEY (CAMP_ID, ITEM_ID),
	FOREIGN KEY (ITEM_ID) REFERENCES ITEM(ITEM_ID) ON DELETE
CASCADE,
	FOREIGN KEY (CAMP_ID) REFERENCES DONATION_CAMP(CAMP_ID) ON
DELETE CASCADE
);

CREATE TABLE TEAM (
	CAMP_ID INT,
	STAFF_ID INT,
	FOREIGN KEY (CAMP_ID) REFERENCES DONATION_CAMP(CAMP_ID) ON
DELETE CASCADE,
```

```
        FOREIGN KEY (STAFF_ID) REFERENCES STAFF(STAFF_ID) ON DELETE
CASCADE,
        PRIMARY KEY (CAMP_ID, STAFF_ID)
);

CREATE TABLE BLOOD_PACKET (
        PACKET_ID SERIAL PRIMARY KEY,
        CAPACITY INT,
        CAMP_ID INT,
        DONOR_ID INT,
        USAGE_TYPE VARCHAR(10),
        FOREIGN KEY (CAMP_ID) REFERENCES DONATION_CAMP(CAMP_ID) ON
DELETE CASCADE,
        FOREIGN KEY (DONOR_ID) REFERENCES DONOR(DONOR_ID) ON DELETE
CASCADE
);

CREATE TABLE BLOOD_REPORT (
        REPORT_ID SERIAL PRIMARY KEY,
        PACKET_ID INT,
        BLOOD_GROUP VARCHAR(4),
        RBC_COUNT NUMERIC(10, 2),
        WBC_COUNT INT,
        PLATELETS VARCHAR(10),
        SUGAR_LEVEL VARCHAR(10),
        HAEMOGLOBIN VARCHAR(10),
        TIME_STAMP TIMESTAMP,
        FOREIGN KEY (PACKET_ID) REFERENCES BLOOD_PACKET(PACKET_ID) ON
DELETE CASCADE
);

CREATE TABLE TRANSFUSION (
        PATIENT_ID INT,
        PACKET_ID INT,
        DOCTOR_ID INT,
        TIME_STAMP TIMESTAMP,
        PRIMARY KEY (DOCTOR_ID, TIME_STAMP),
        FOREIGN KEY (PACKET_ID) REFERENCES BLOOD_PACKET(PACKET_ID) ON
DELETE CASCADE,
```

```
        FOREIGN KEY (PATIENT_ID) REFERENCES PATIENT(PATIENT_ID) ON
DELETE CASCADE,
        FOREIGN KEY (DOCTOR_ID) REFERENCES DOCTOR(DOCTOR_ID) ON
DELETE CASCADE
);
```

Insertion Queries:

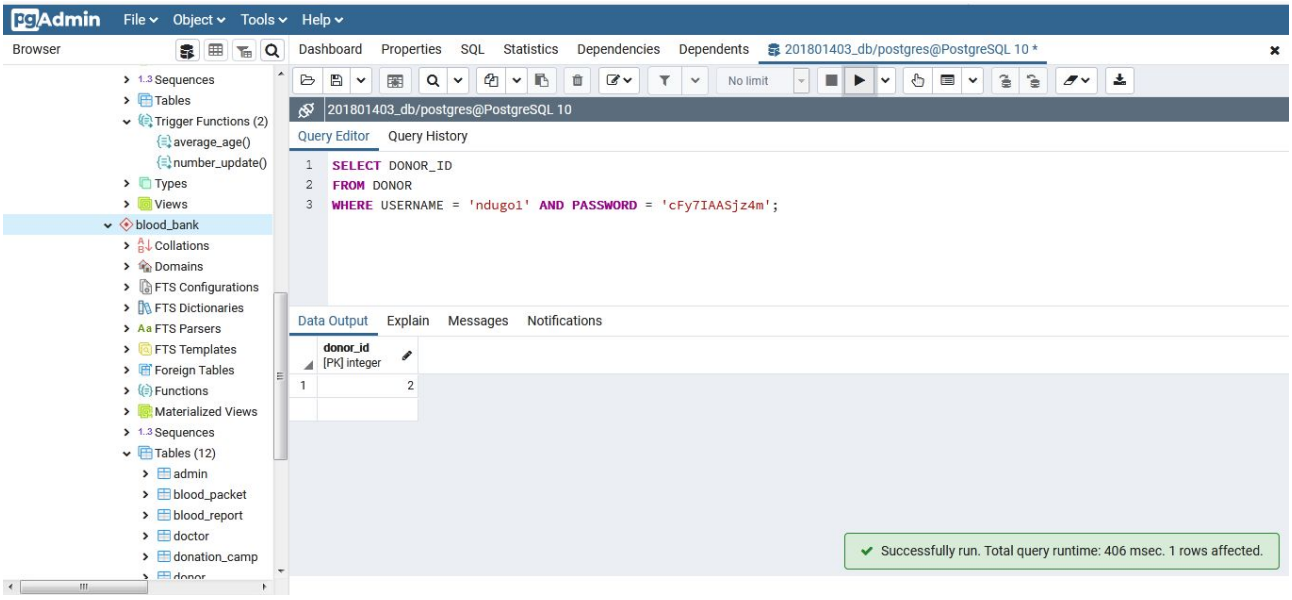
```
COPY PUBLIC.ADMIN(NAME, ADDRESS, MOBILE, USERNAME, PASSWORD) FROM
'd:/bloodbank/Admin.csv' DELIMITER ',' CSV HEADER;

COPY PUBLIC.DONOR(USERNAME, PASSWORD, NAME, ADDRESS, MOBILE) FROM
'd:/bloodbank/Donor.csv' DELIMITER ',' CSV HEADER;

COPY PUBLIC.DOCTOR(USERNAME, PASSWORD, NAME, ADDRESS, MOBILE)
FROM 'd:/bloodbank/Doctor.csv' DELIMITER ',' CSV HEADER;

COPY PUBLIC.PATIENT(NAME, ADDRESS, MOBILE) FROM 'd:/bloodbank/Patient.csv'
DELIMITER ',' CSV HEADER;

COPY PUBLIC.STAFF(username, password, position, date_of_joining, name, address,
mobile, salary) FROM 'd:/bloodbank/Staff.csv' DELIMITER ',' CSV HEADER;

COPY PUBLIC.ITEM(name, type, item_count, price) FROM 'd:/bloodbank/Item.csv'
DELIMITER ',' CSV HEADER;

COPY PUBLIC.DONATION_CAMP(event_date, location, start_time, end_time) FROM
'd:/bloodbank/Camp.csv' DELIMITER ',' CSV HEADER;

COPY PUBLIC.INVENTORY(camp_id, item_id, count) FROM 'd:/bloodbank/Inventory.csv'
DELIMITER ',' CSV HEADER;

COPY PUBLIC.TEAM(staff_id, camp_id) FROM 'd:/bloodbank/Team.csv' DELIMITER ','
CSV HEADER;

COPY PUBLIC.BLOOD_PACKET(capacity, camp_id, donor_id, usage_type) FROM
'd:/bloodbank/Packet.csv' DELIMITER ',' CSV HEADER;
```

```
COPY PUBLIC.BLOOD_REPORT(packet_id, platelets, haemoglobin, sugar_level,
wbc_count, rbc_count, blood_group, time_stamp) FROM 'd:/bloodbank/Report.csv'
DELIMITER ',' CSV HEADER;

COPY PUBLIC.TRANSFUSION(packet_id, patient_id, doctor_id,time_stamp) FROM
'd:/bloodbank/Transfusion.csv' DELIMITER ',' CSV HEADER;
```

SQL Queries and Their Snapshots:

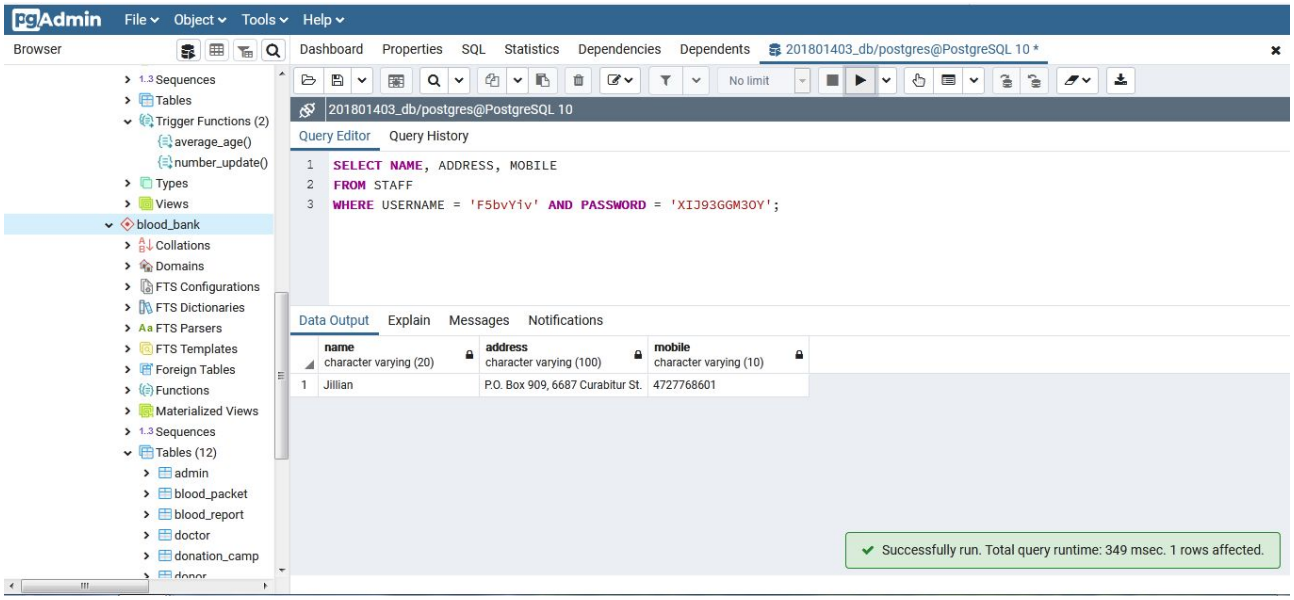
CREDENTIALS:

1. **SELECT** DONOR_ID
   FROM DONOR
   WHERE USERNAME = 'ndugo1' **AND** PASSWORD = 'cFy7IAASjz4m';



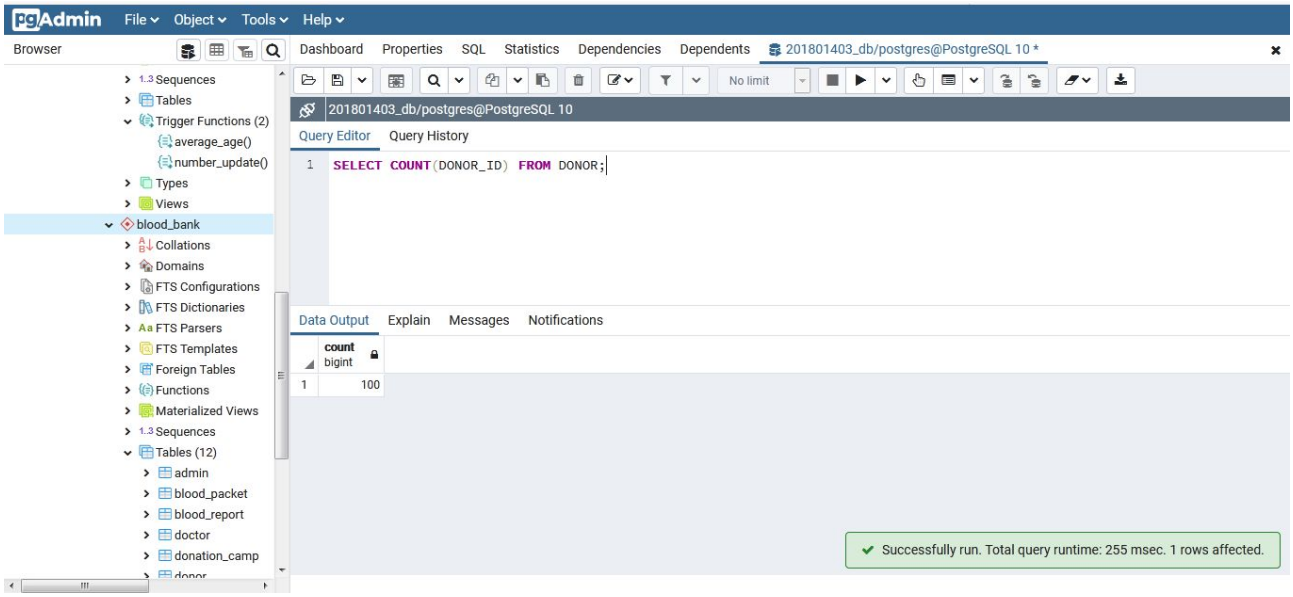
DETAILS:

2. **SELECT** NAME, ADDRESS, MOBILE
   FROM STAFF
   WHERE USERNAME = 'F5bvYiv' AND PASSWORD = 'XIJ93GGM3OY';



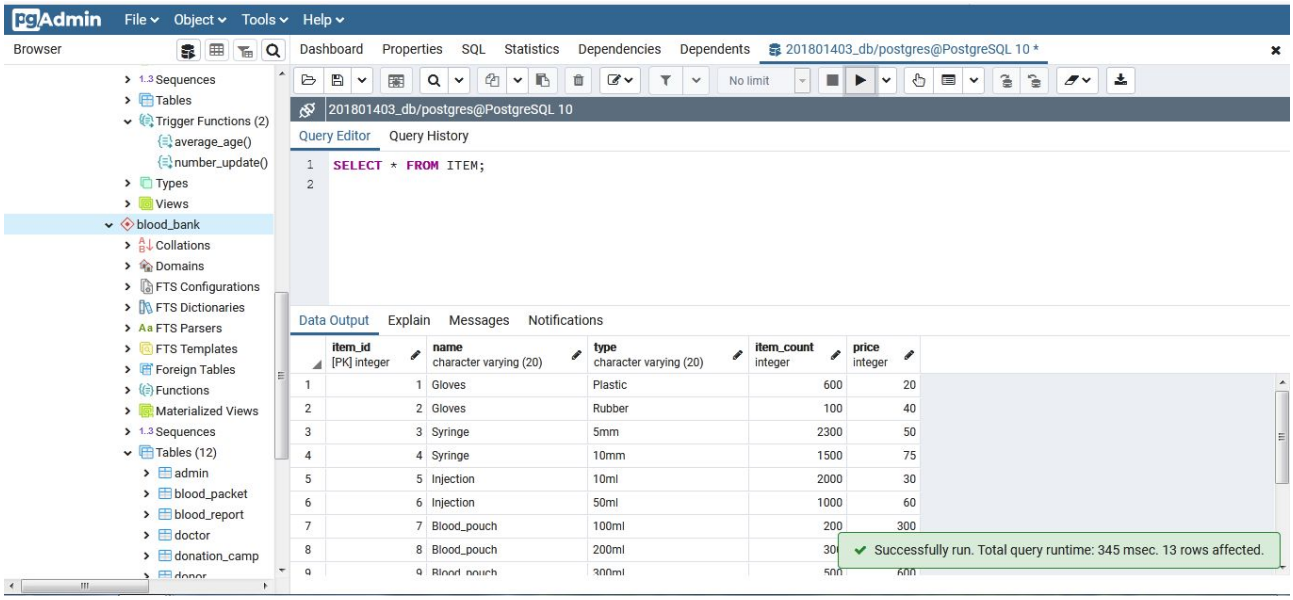
DONOR COUNT: (TOTAL COUNT OF DONORS)

3. **SELECT COUNT(**DONOR_ID**) FROM** DONOR;



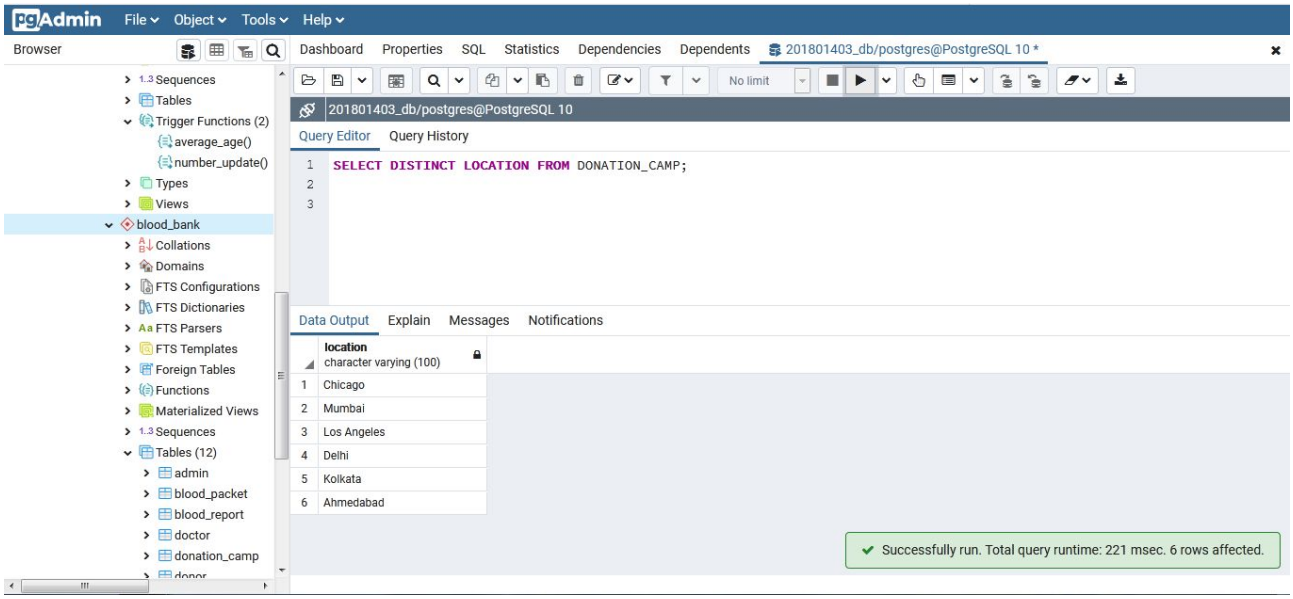
STOCK DETAILS: (DETAILS OF ALL STOCK AVAILABLE AT THE BLOOD BANK)

4. **SELECT * FROM** ITEM;



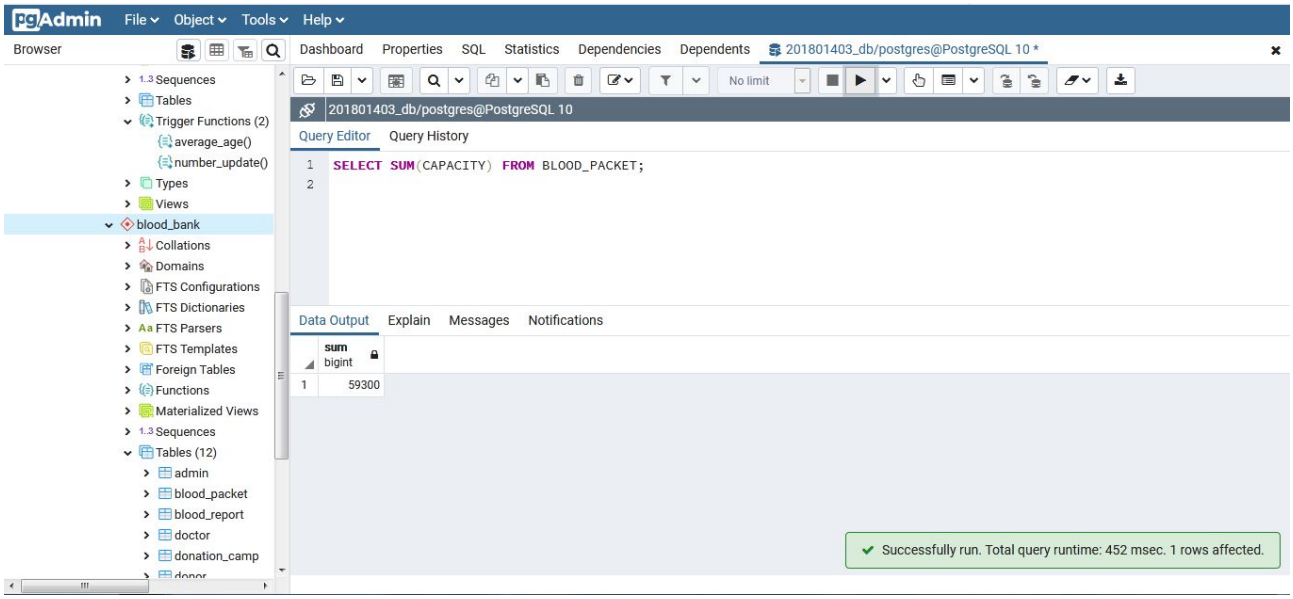
DONATION CAMP ARRANGED:(DETAILS OF ALL THE CAMPS CONDUCTED TILL NOW)

5. **SELECT DISTINCT** LOCATION **FROM** DONATION_CAMP;



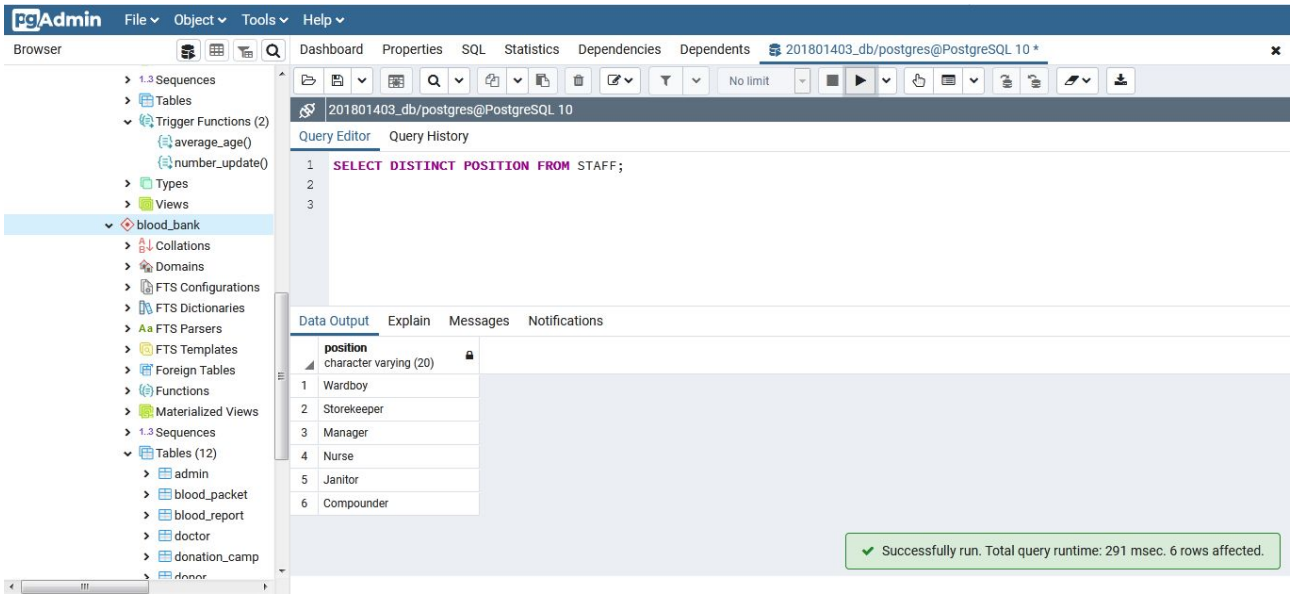
TOTAL BLOOD AVAILABLE AT BANK:

6. **SELECT SUM(**CAPACITY**) FROM** BLOOD_PACKET;



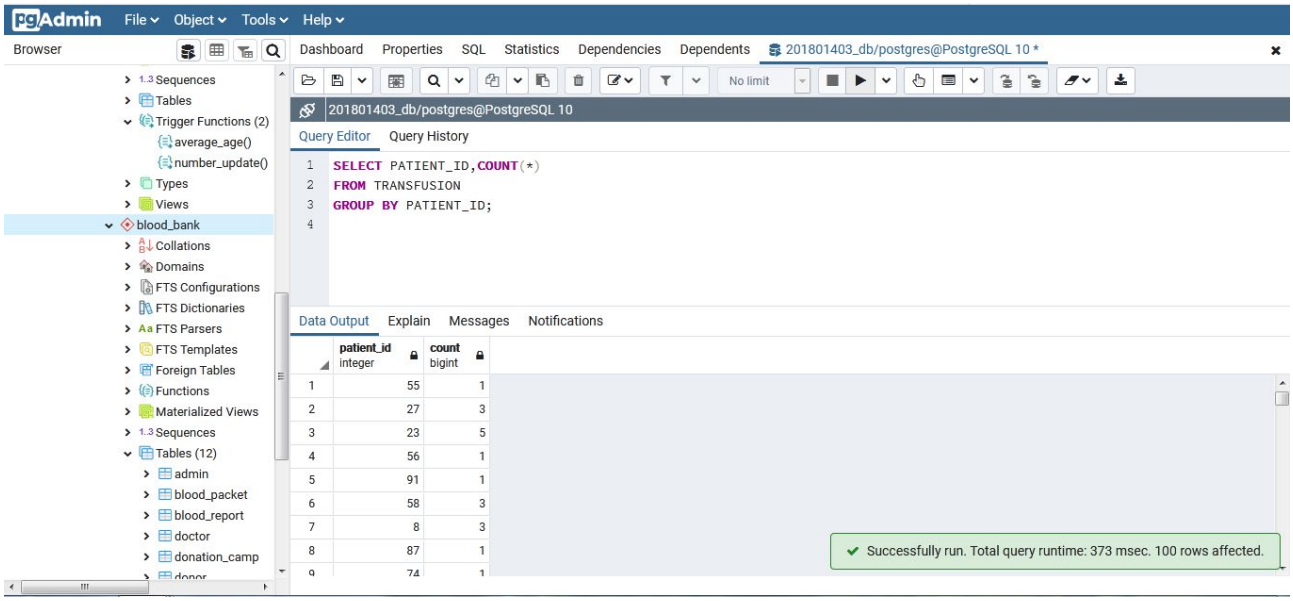
POSITIONS IN THE STAFF:(DIFFERENT POSITIONS AVAILABLE AT THE BLOOD BANK)

7. **SELECT DISTINCT** POSITION **FROM** STAFF;



HOW MANY TIMES A PATIENT NEEDED BLOOD:

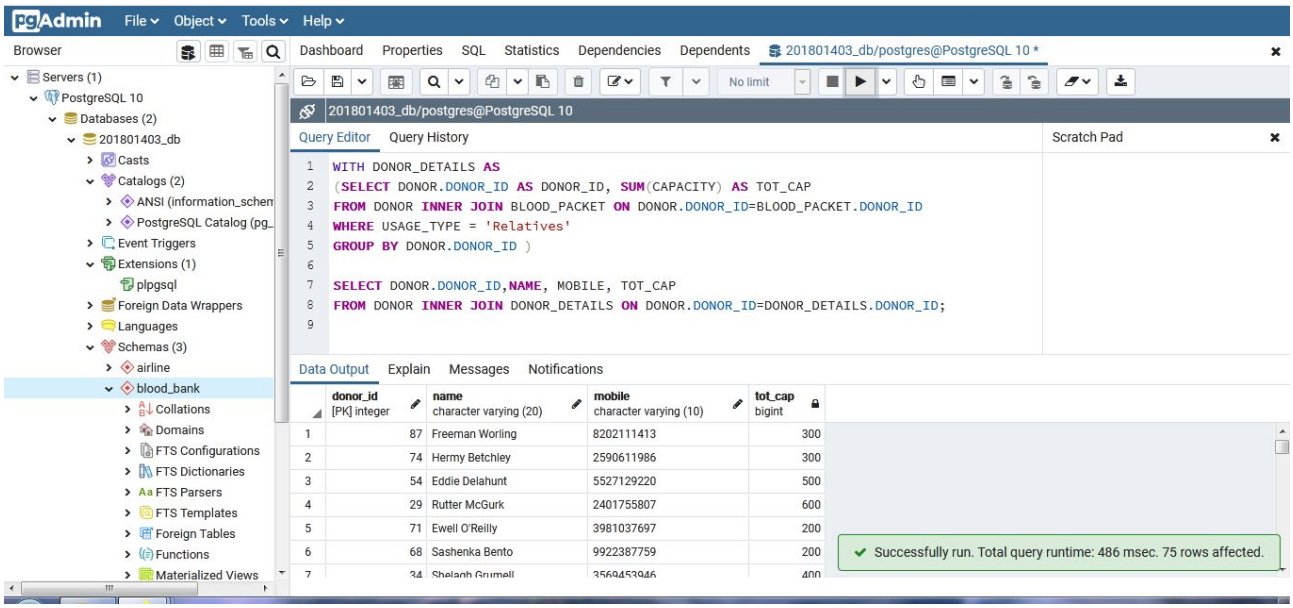
8. **SELECT** PATIENT_ID**,COUNT(*)**
   FROM TRANSFUSION
   GROUP BY PATIENT_ID;



GET THE CAPACITIES OF BLOOD DONATED BY EACH DONOR HAVING USAGE_TYPE EQUAL TO RELATIVES:

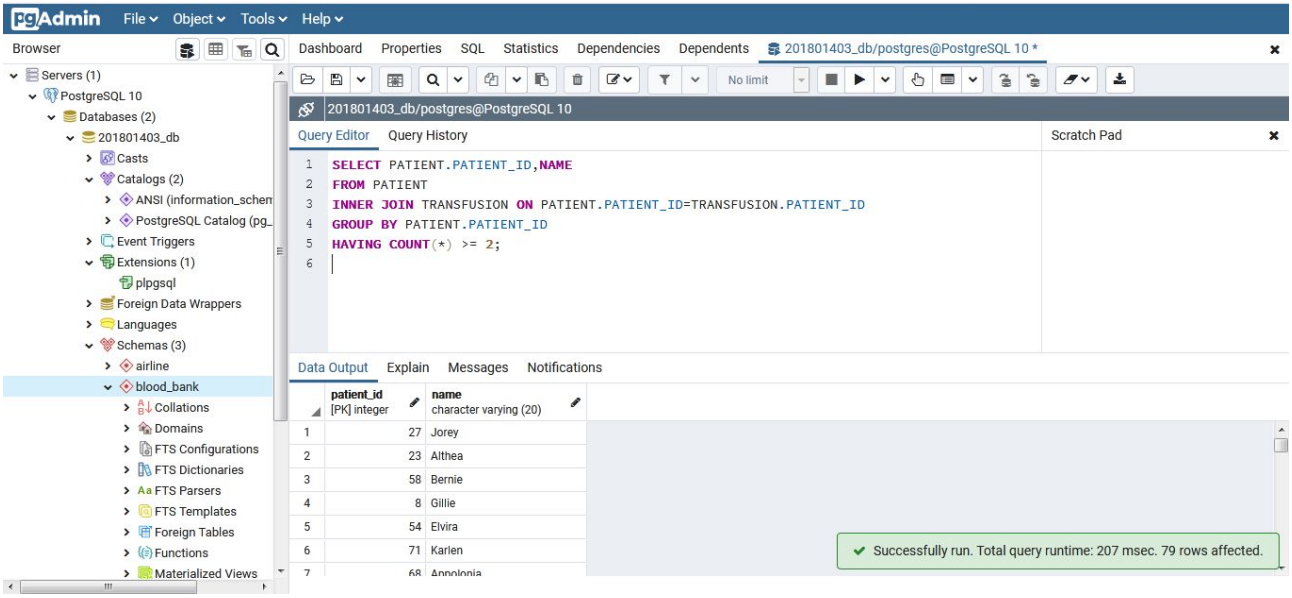
9. WITH DONOR_DETAILS AS
   (SELECT DONOR.DONOR_ID AS DONOR_ID, SUM(CAPACITY) AS TOT_CAP
   FROM DONOR INNER JOIN BLOOD_PACKET ON
   DONOR.DONOR_ID=BLOOD_PACKET.DONOR_ID
   WHERE USAGE_TYPE = 'Relatives'
   GROUP BY DONOR.DONOR_ID)

   SELECT DONOR.DONOR_ID,NAME, MOBILE, TOT_CAP
   FROM DONOR INNER JOIN DONOR_DETAILS ON
   DONOR.DONOR_ID=DONOR_DETAILS.DONOR_ID;



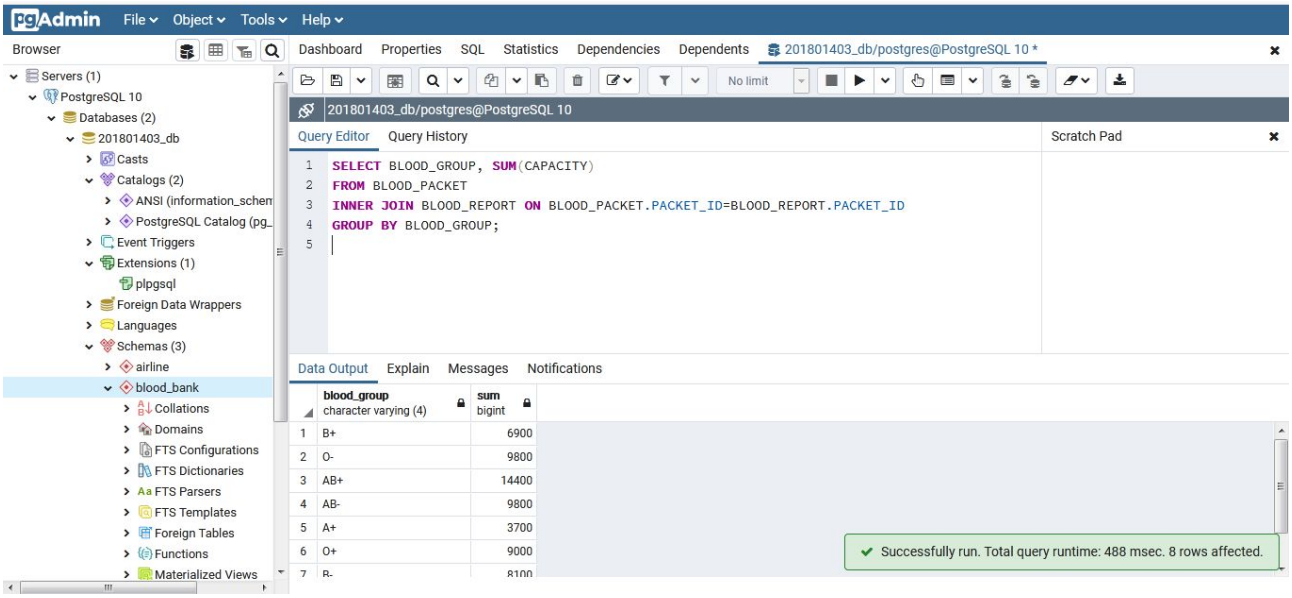
THIS QUERY RETURNS ALL THE NAMES OF THE PATIENTS WHO HAVE RECEIVED BLOOD MORE THAN EQUAL TO 2.

10. SELECT PATIENT.PATIENT_ID,NAME
    FROM PATIENT
    INNER JOIN TRANSFUSION ON PATIENT.PATIENT_ID=TRANSFUSION.PATIENT_ID
    GROUP BY PATIENT.PATIENT_ID
    HAVING COUNT(*) >= 2;



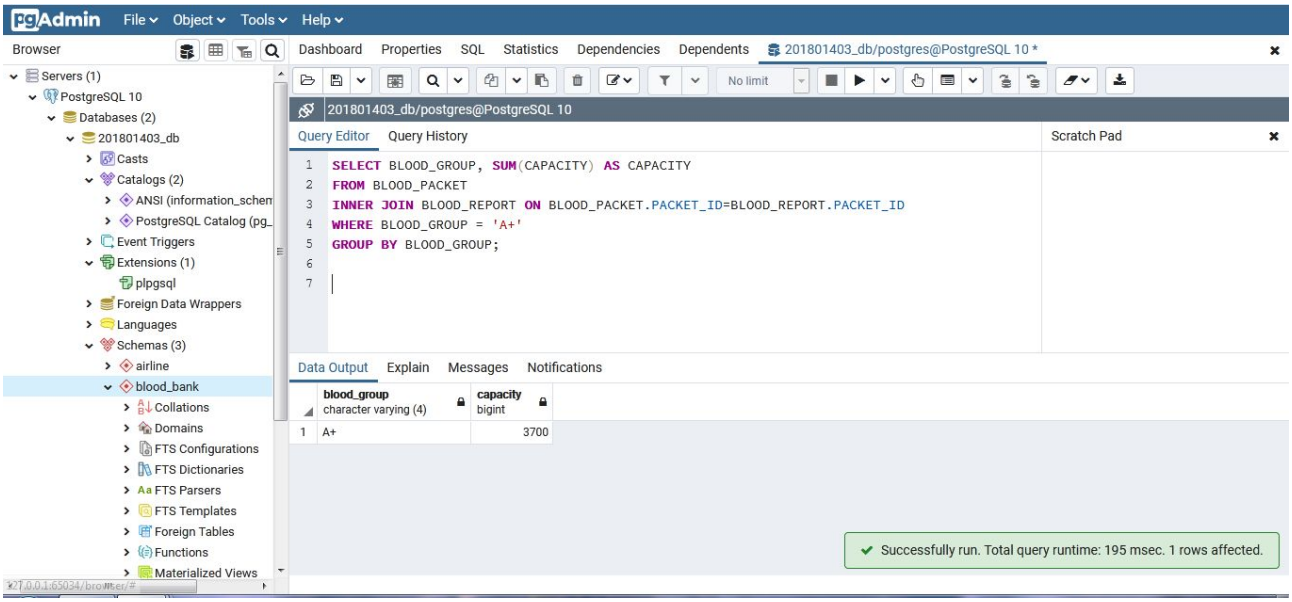
GET THE TOTAL COLLECTED BLOOD HAVING DIFFERENT BLOOD GROUPS

11. SELECT BLOOD_GROUP, SUM(CAPACITY)
    FROM BLOOD_PACKET
    INNER JOIN BLOOD_REPORT ON
    BLOOD_PACKET.PACKET_ID=BLOOD_REPORT.PACKET_ID
    GROUP BY BLOOD_GROUP;



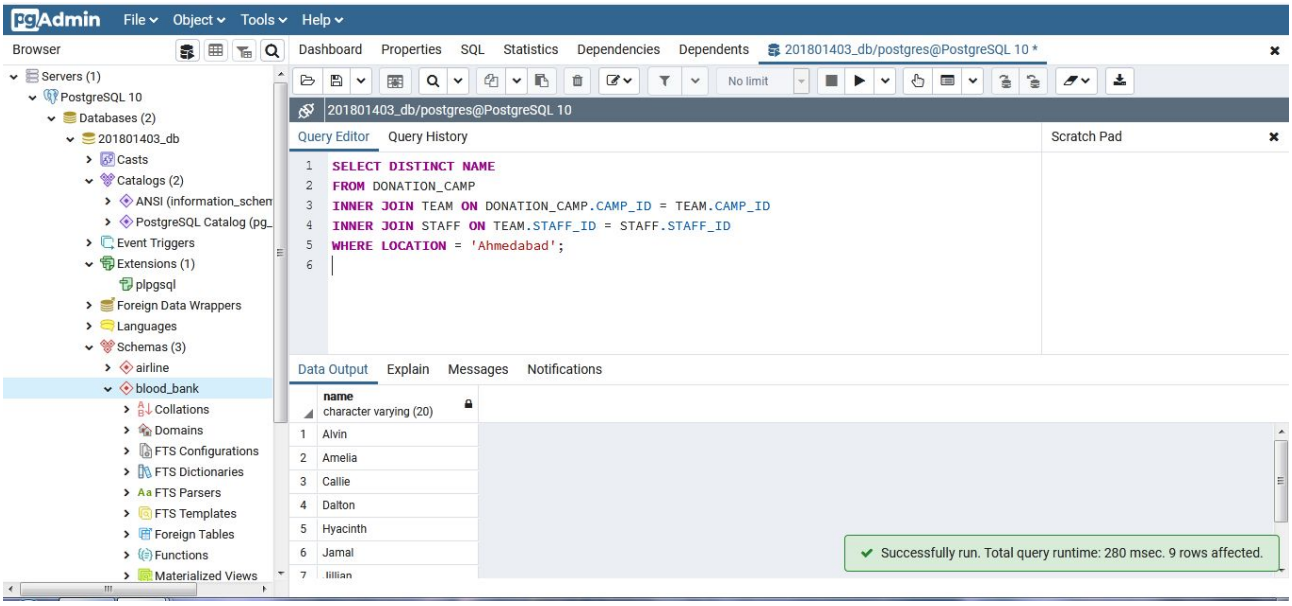
SPECIFIC BLOOD GROUP CAPACITY (i.e., A+)

12. SELECT BLOOD_GROUP, SUM(CAPACITY) AS CAPACITY
    FROM BLOOD_PACKET
    INNER JOIN BLOOD_REPORT ON
    BLOOD_PACKET.PACKET_ID=BLOOD_REPORT.PACKET_ID
    WHERE BLOOD_GROUP = 'A+'
    GROUP BY BLOOD_GROUP;



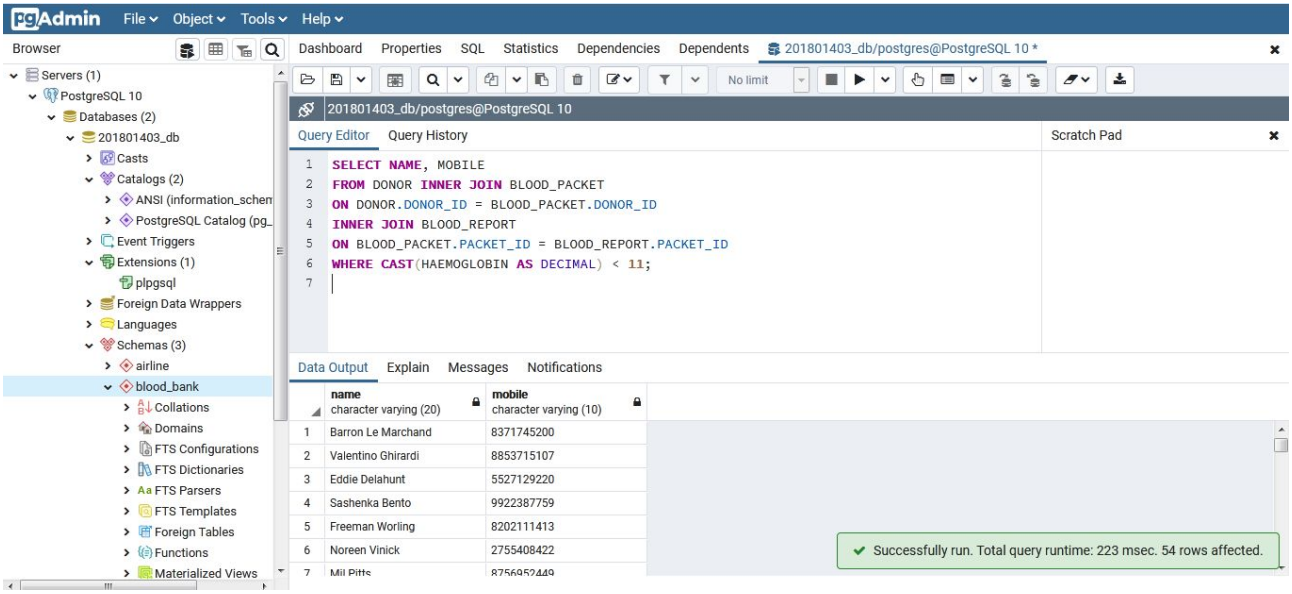
NAME OF ALL THE STAFF MEMBERS WHICH ARE PART OF THE CAMPS CONDUCTED IN AHMEDABAD

13. SELECT DISTINCT NAME
    FROM DONATION_CAMP
    INNER JOIN TEAM ON DONATION_CAMP.CAMP_ID = TEAM.CAMP_ID
    INNER JOIN STAFF ON TEAM.STAFF_ID = STAFF.STAFF_ID
    WHERE LOCATION = 'Ahmedabad';



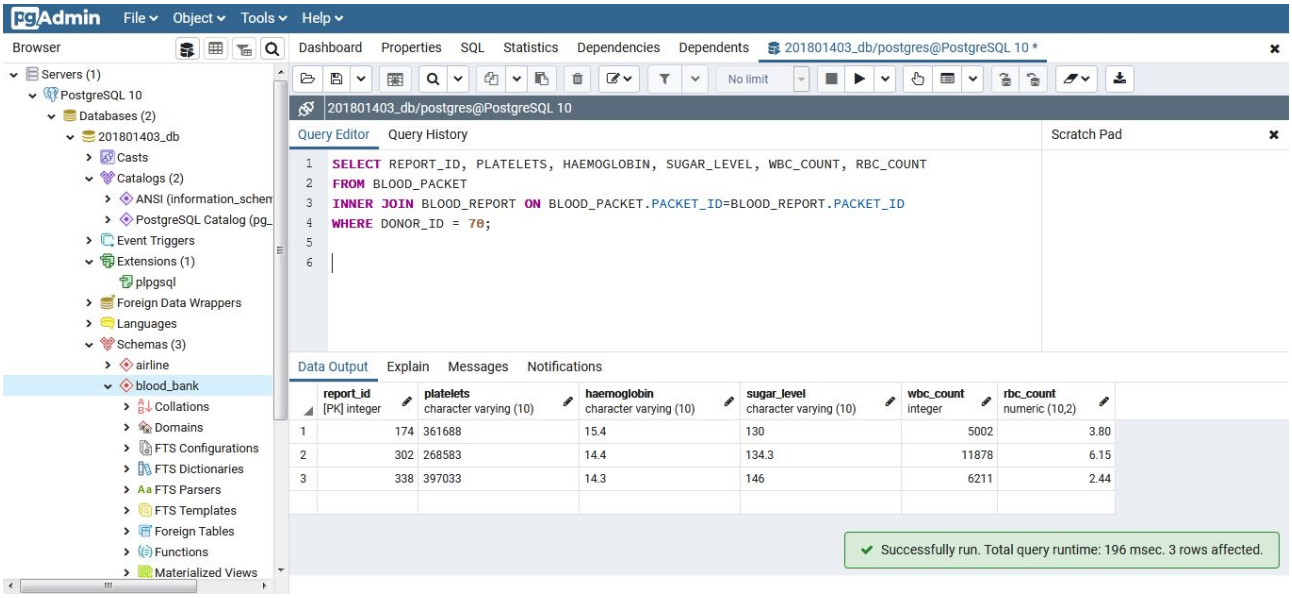
GET THE NAMES AND CONTACT NUMBER OF DONORS HAVING LESS THAN 11 HEMOGLOBIN LEVEL

14. SELECT NAME, MOBILE
    FROM DONOR INNER JOIN BLOOD_PACKET
    ON DONOR.DONOR_ID = BLOOD_PACKET.DONOR_ID
    INNER JOIN BLOOD_REPORT
    ON BLOOD_PACKET.PACKET_ID = BLOOD_REPORT.PACKET_ID
    WHERE CAST(HAEMOGLOBIN AS DECIMAL) < 11;



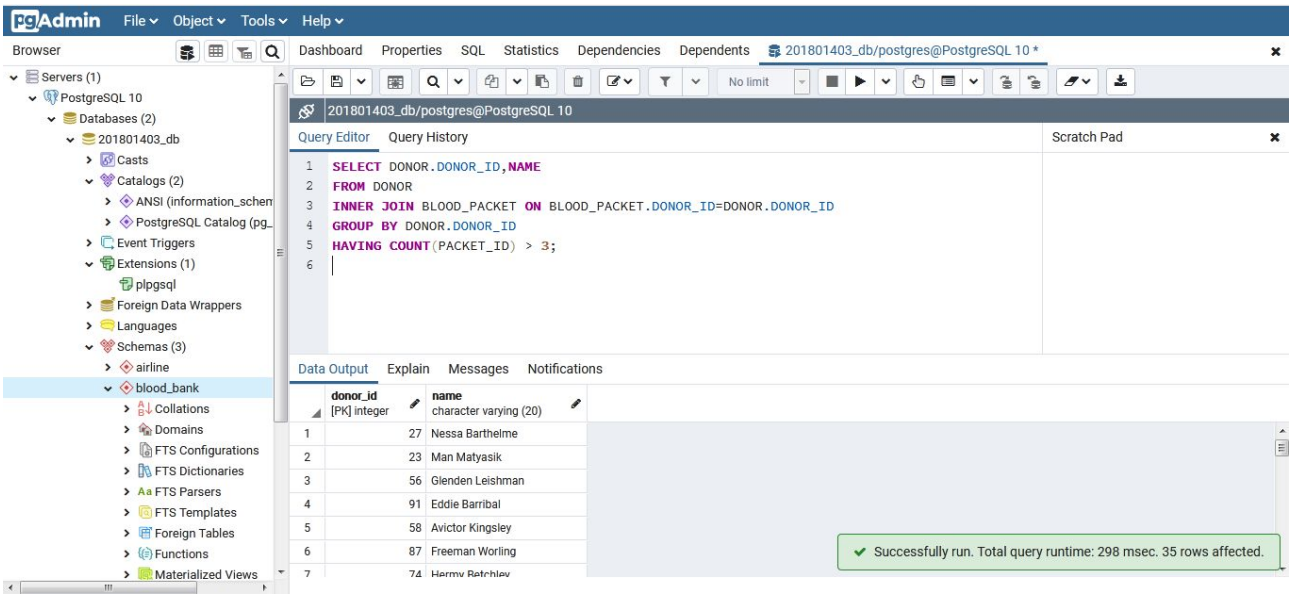
TO GET THE DETAILS OF ALL THE BLOOD DONATION MADE BY PARTICULAR PERSON:

15. SELECT REPORT_ID, PLATELETS, HAEMOGLOBIN, SUGAR_LEVEL, WBC_COUNT, RBC_COUNT
    FROM BLOOD_PACKET
    INNER JOIN BLOOD_REPORT ON
    BLOOD_PACKET.PACKET_ID=BLOOD_REPORT.PACKET_ID
    WHERE DONOR_ID = 70;



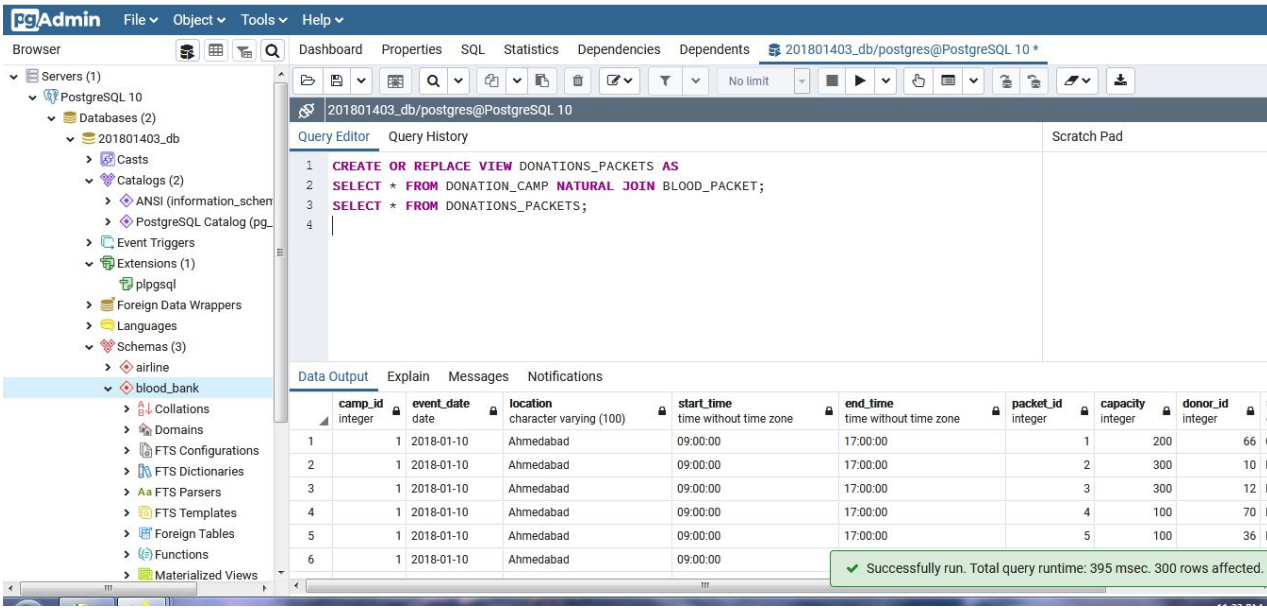
TO GET ALL THE DONORS WHO HAVE DONATED MORE THAN 3 TIMES:

16. SELECT DONOR.DONOR_ID,NAME
    FROM DONOR
    INNER JOIN BLOOD_PACKET ON BLOOD_PACKET.DONOR_ID=DONOR.DONOR_ID
    GROUP BY DONOR.DONOR_ID
    HAVING COUNT(PACKET_ID) > 3;



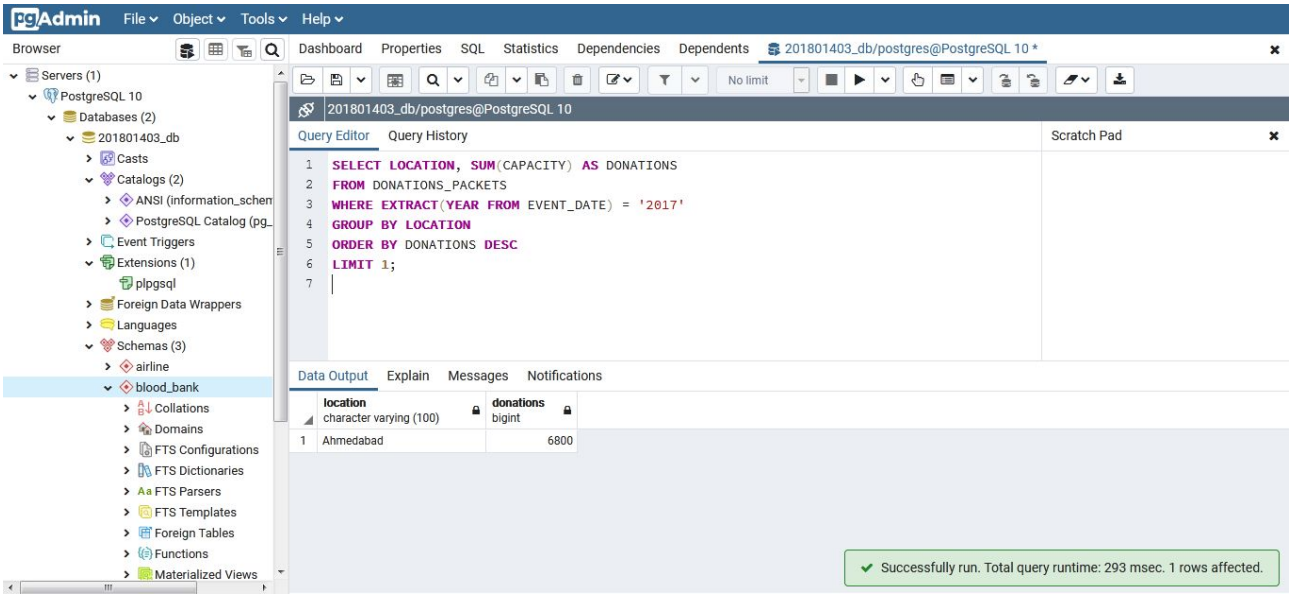
CITY RANKING

17. CREATE OR REPLACE VIEW DONATIONS_PACKETS AS
    SELECT * FROM DONATION_CAMP NATURAL JOIN BLOOD_PACKET;
    SELECT * FROM DONATIONS_PACKETS;



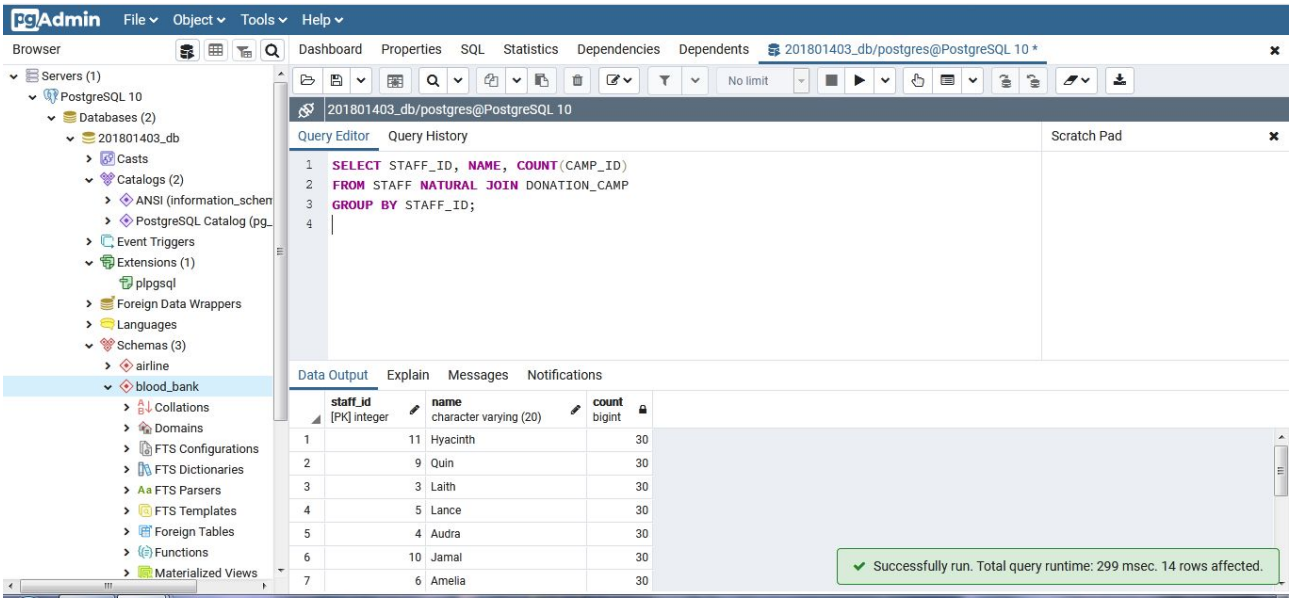
TOP CITY IN 2017 IN TERMS OF DONATION:

18. SELECT LOCATION, SUM(CAPACITY) AS DONATIONS
    FROM DONATIONS_PACKETS
    WHERE EXTRACT(YEAR FROM EVENT_DATE) = '2017'
    GROUP BY LOCATION
    ORDER BY DONATIONS DESC
    LIMIT 1;



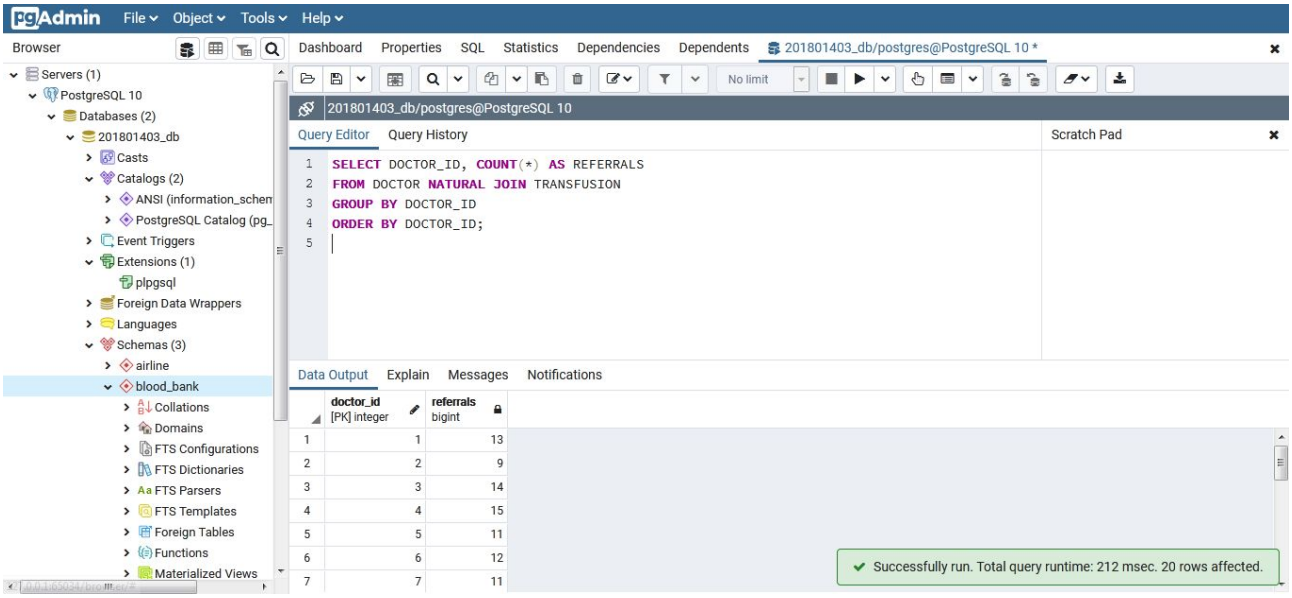
GET THE DETAILS OF STAFF MEMBERS WITH THEIR CONTRIBUTION IN DIFFERENT BLOOD DONATION CAMP

19. SELECT STAFF_ID, NAME, COUNT(CAMP_ID)
    FROM STAFF NATURAL JOIN DONATION_CAMP
    GROUP BY STAFF_ID;



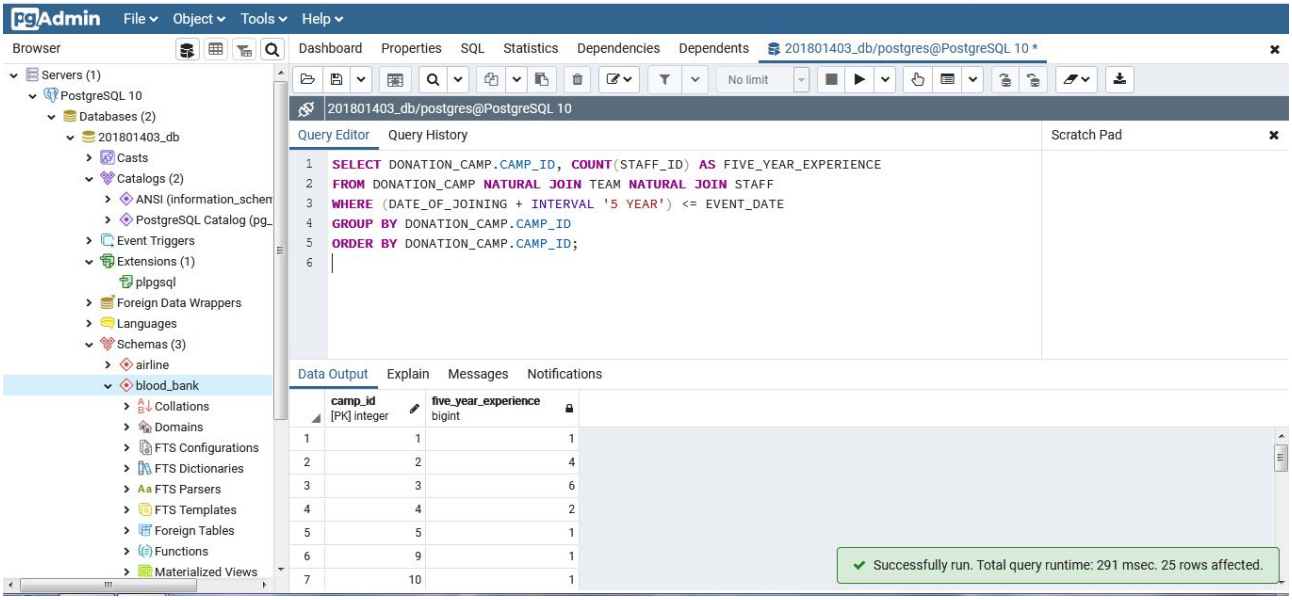
GET THE DETAILS AND COUNT OF ALL DOCTORS WHO HAVE REQUESTED FOR BLOOD:

20. SELECT DOCTOR_ID, COUNT(*) AS REFERRALS
    FROM DOCTOR NATURAL JOIN TRANSFUSION
    GROUP BY DOCTOR_ID
    ORDER BY DOCTOR_ID;



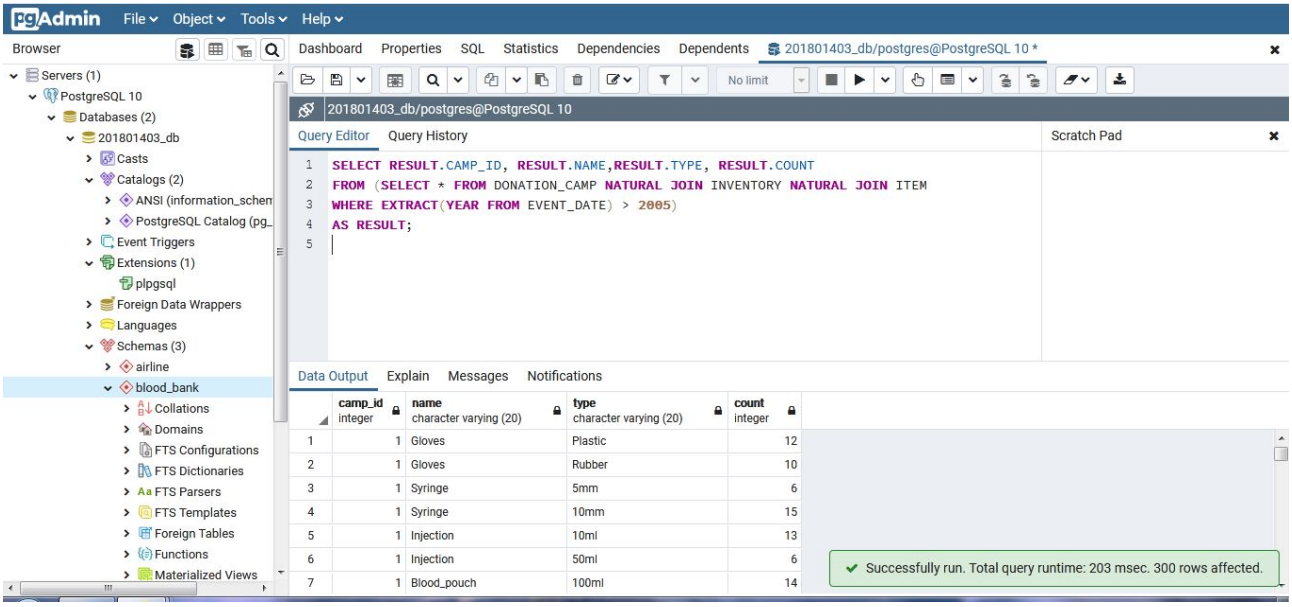
FOR ALL CAMP, RETURN NUMBER OF STAFF MEMBER HAVING MORE THAN 5 YEARS OF EXPERIENCE

21. SELECT DONATION_CAMP.CAMP_ID, COUNT(STAFF_ID) AS FIVE_YEAR_EXPERIENCE
    FROM DONATION_CAMP NATURAL JOIN TEAM NATURAL JOIN STAFF
    WHERE (DATE_OF_JOINING + INTERVAL '5 YEAR') <= EVENT_DATE
    GROUP BY DONATION_CAMP.CAMP_ID
    ORDER BY DONATION_CAMP.CAMP_ID;



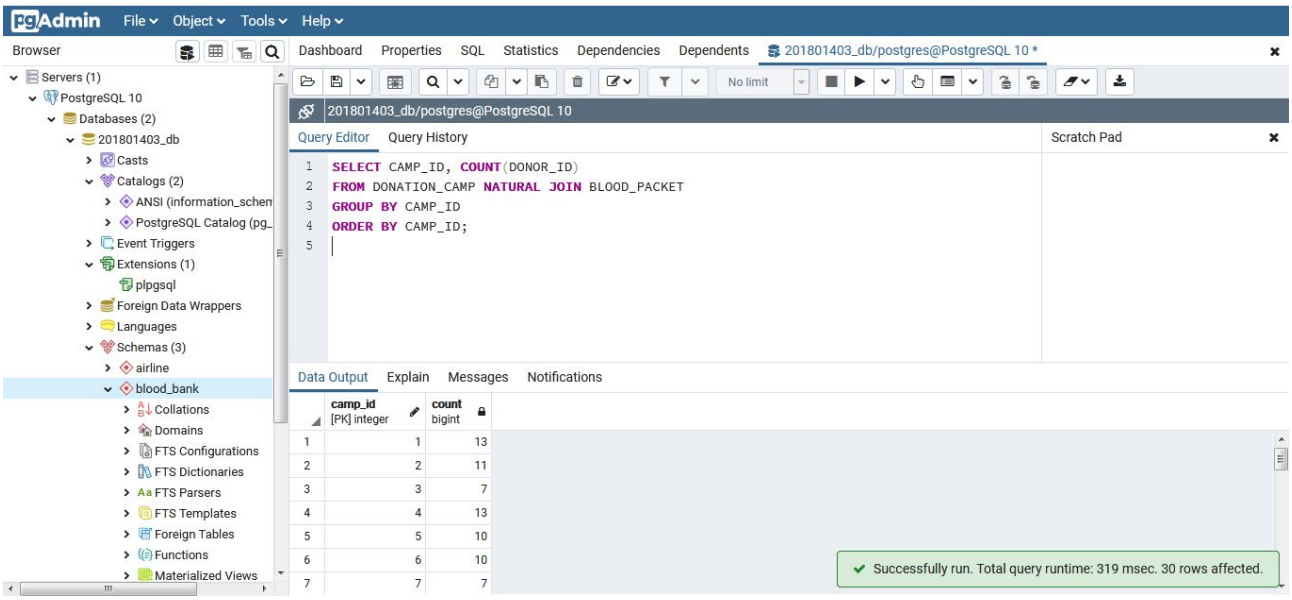
GET ALL THE DETAILS OF INVENTORY USED IN CAMPS CONDUCTED AFTER 2020:

22. SELECT RESULT.CAMP_ID, RESULT.NAME,RESULT.TYPE, RESULT.COUNT
FROM (SELECT * FROM DONATION_CAMP NATURAL JOIN INVENTORY NATURAL JOIN ITEM
WHERE EXTRACT(YEAR FROM EVENT_DATE) > 2005)
AS RESULT;



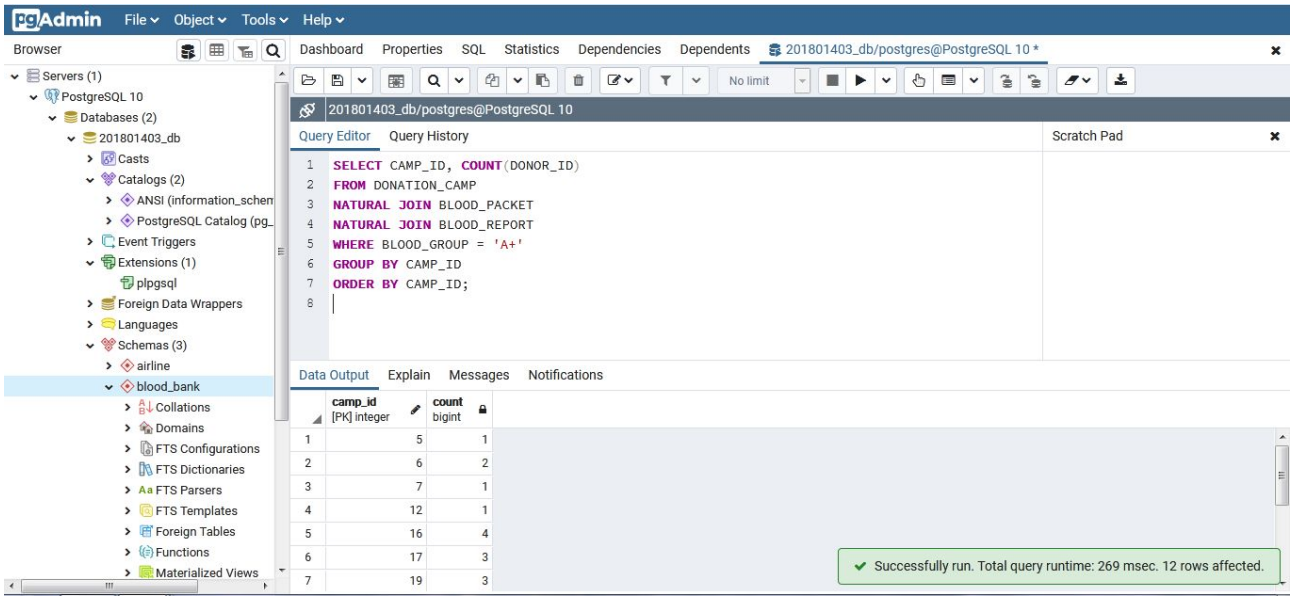
GET THE COUNT OF DONORS IN EACH DONATION CAMP:

23. SELECT CAMP_ID, COUNT(DONOR_ID)
    FROM DONATION_CAMP NATURAL JOIN BLOOD_PACKET
    GROUP BY CAMP_ID
    ORDER BY CAMP_ID;



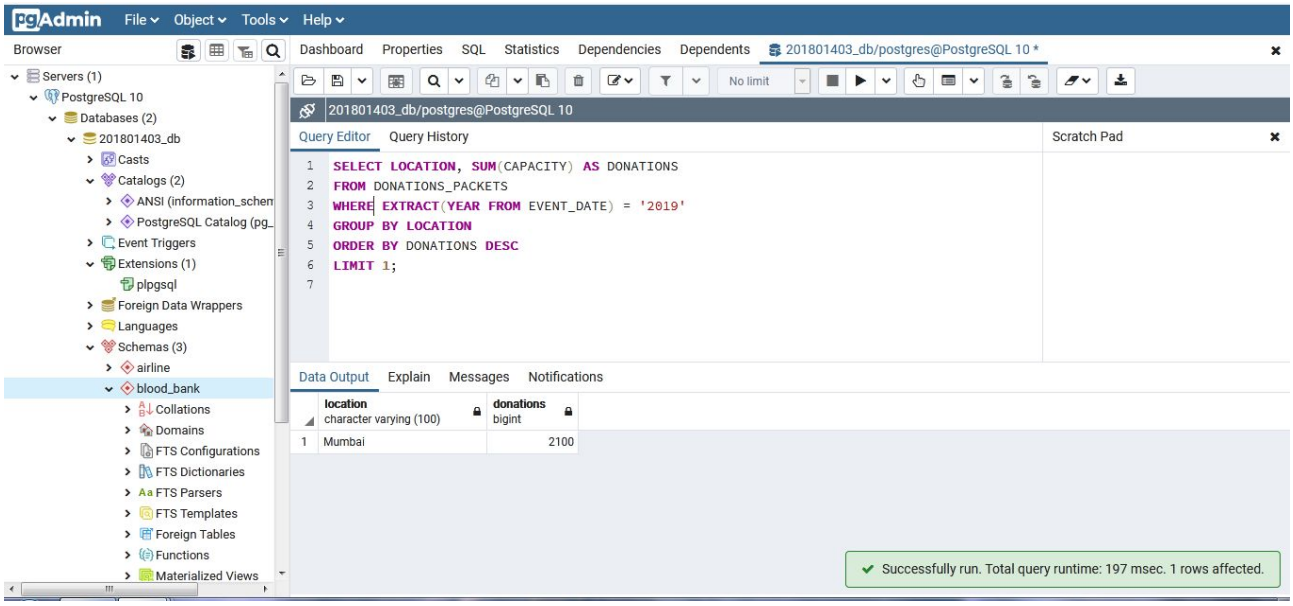
GET THE COUNT OF DONORS OF ‘A+’ IN EACH DONATION CAMP:

24. SELECT CAMP_ID, COUNT(DONOR_ID)
    FROM DONATION_CAMP
    NATURAL JOIN BLOOD_PACKET
    NATURAL JOIN BLOOD_REPORT
    WHERE BLOOD_GROUP = 'A+'
    GROUP BY CAMP_ID
    ORDER BY CAMP_ID;



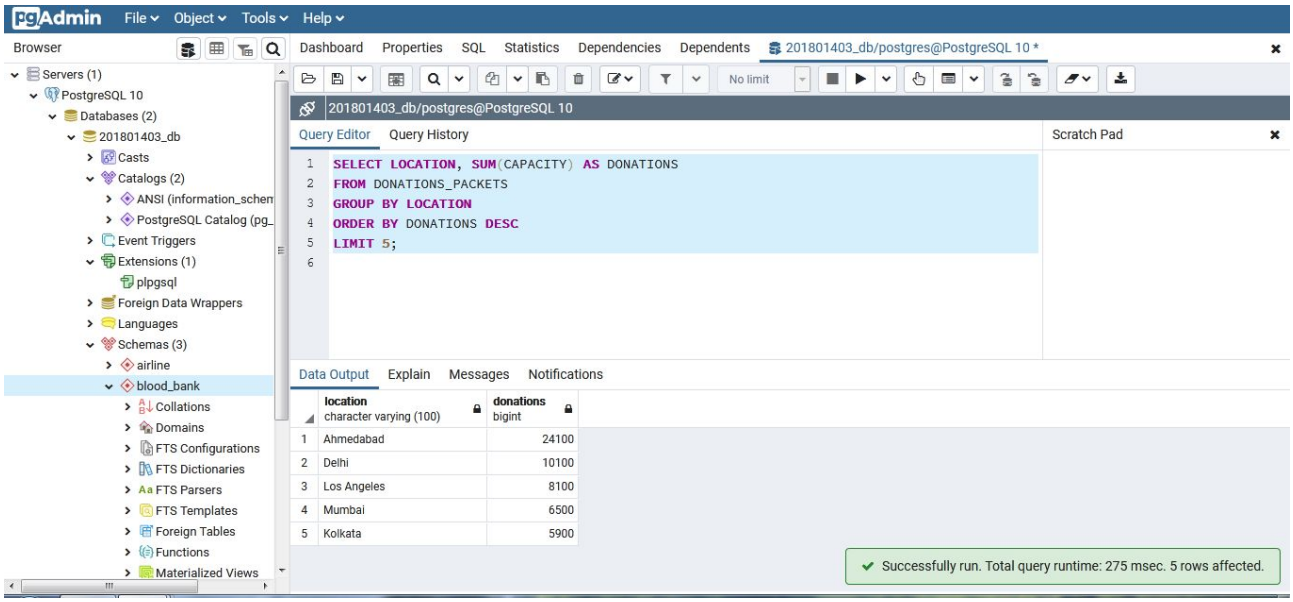
TOP CITY IN 2019 IN TERMS OF DONATION:

25. SELECT LOCATION, SUM(CAPACITY) AS DONATIONS
    FROM DONATIONS_PACKETS
    WHERE EXTRACT(YEAR FROM EVENT_DATE) = '2019'
    GROUP BY LOCATION
    ORDER BY DONATIONS DESC
    LIMIT 1;



TOP 5 CITIES IN DONATION:

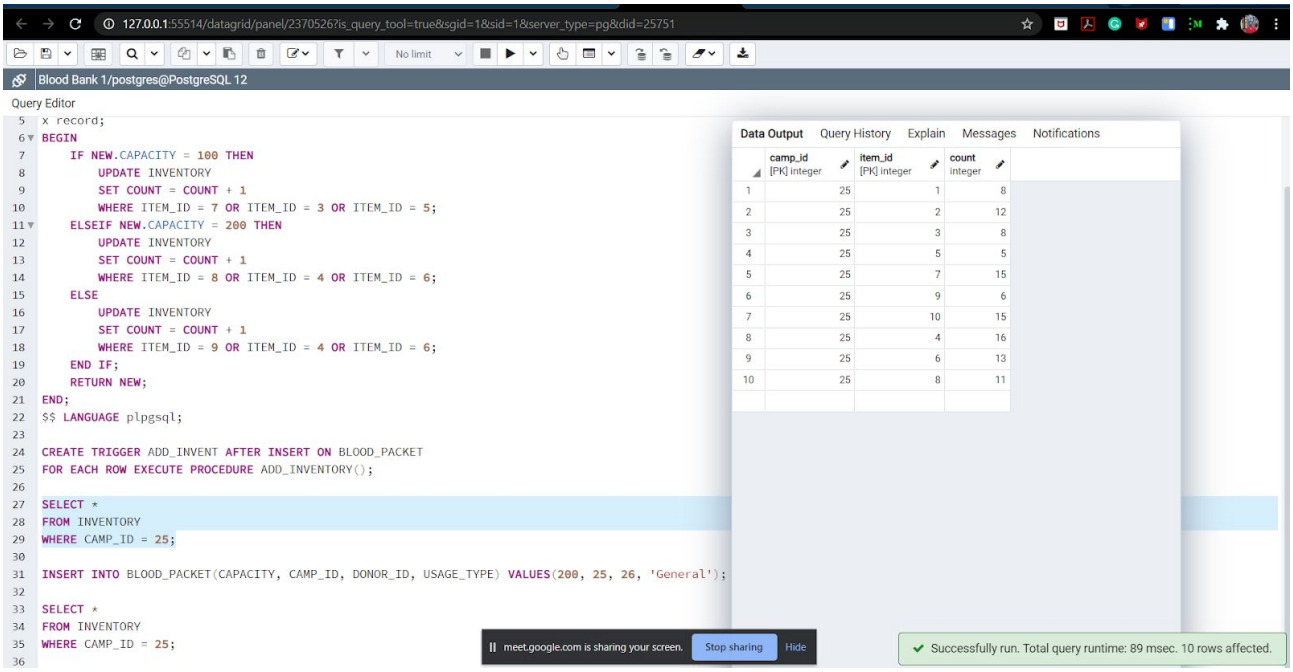
26. **SELECT** LOCATION, **SUM(**CAPACITY**) AS** DONATIONS
    FROM DONATIONS_PACKETS
    GROUP BY LOCATION
    SORT BY DONATIONS **DESC**
    LIMIT 5;



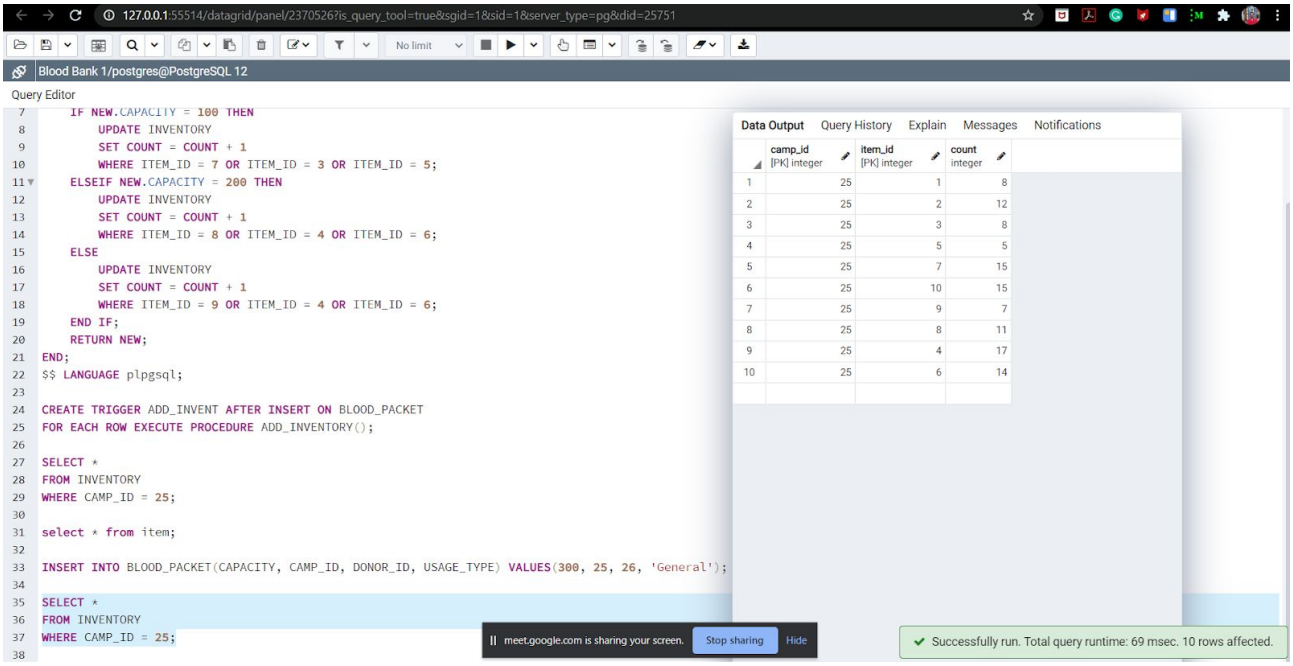
A Trigger to update the inventory when a donor donates and items are used

27.
```
CREATE OR REPLACE FUNCTION ADD_INVENTORY()
RETURNS TRIGGER AS
$$
DECLARE
x record;
BEGIN
        IF NEW.CAPACITY = 100 THEN
                UPDATE INVENTORY
                SET COUNT = COUNT + 1
                WHERE ITEM_ID = 7 OR ITEM_ID = 3 OR ITEM_ID = 5;
        ELSEIF NEW.CAPACITY = 200 THEN
                UPDATE INVENTORY
                SET COUNT = COUNT + 1
                WHERE ITEM_ID = 8 OR ITEM_ID = 4 OR ITEM_ID = 6;
        ELSE
                UPDATE INVENTORY
                SET COUNT = COUNT + 1
                WHERE ITEM_ID = 9 OR ITEM_ID = 4 OR ITEM_ID = 6;
        END IF;
        RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER ADD_INVENT AFTER INSERT ON BLOOD_PACKET
FOR EACH ROW EXECUTE PROCEDURE ADD_INVENTORY();

SELECT *
FROM INVENTORY
WHERE CAMP_ID = 25

INSERT INTO BLOOD_PACKET(CAPACITY, CAMP_ID, DONOR_ID, USAGE_TYPE)
VALUES(200, 25, 26, 'General');

SELECT *
FROM INVENTORY
WHERE CAMP_ID = 25
```

Before:

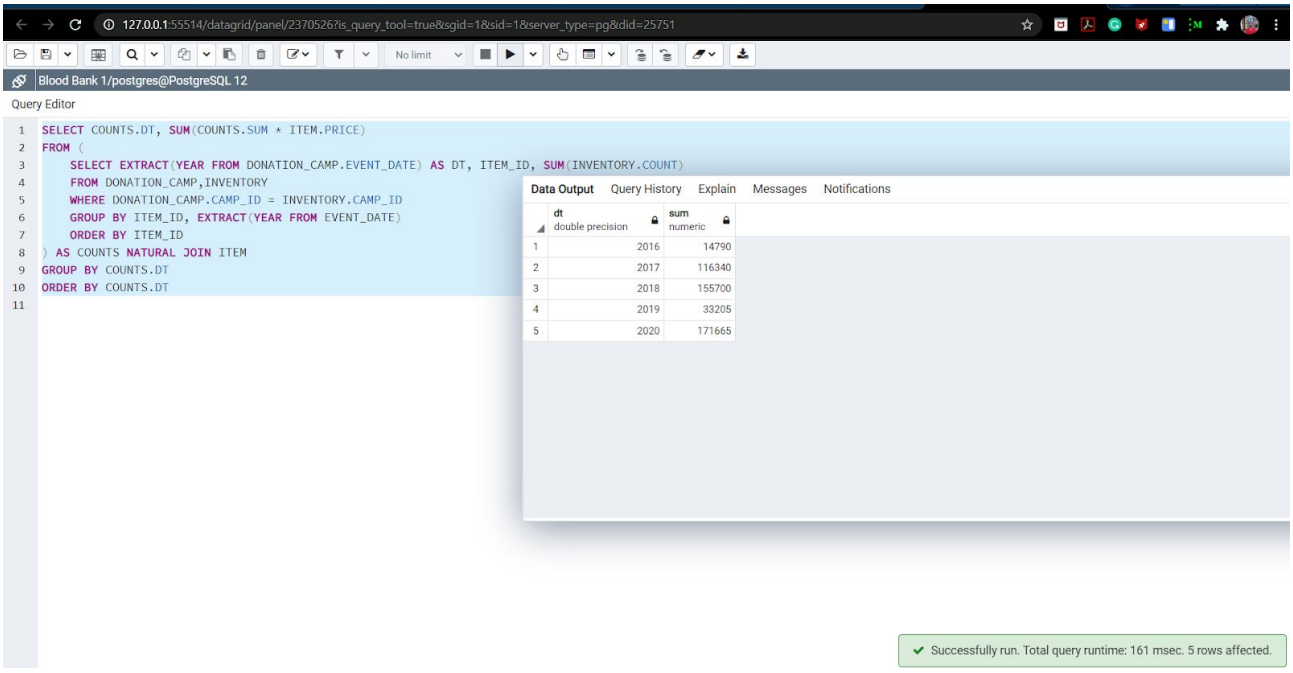


After:



Calculate cost of inventory every year

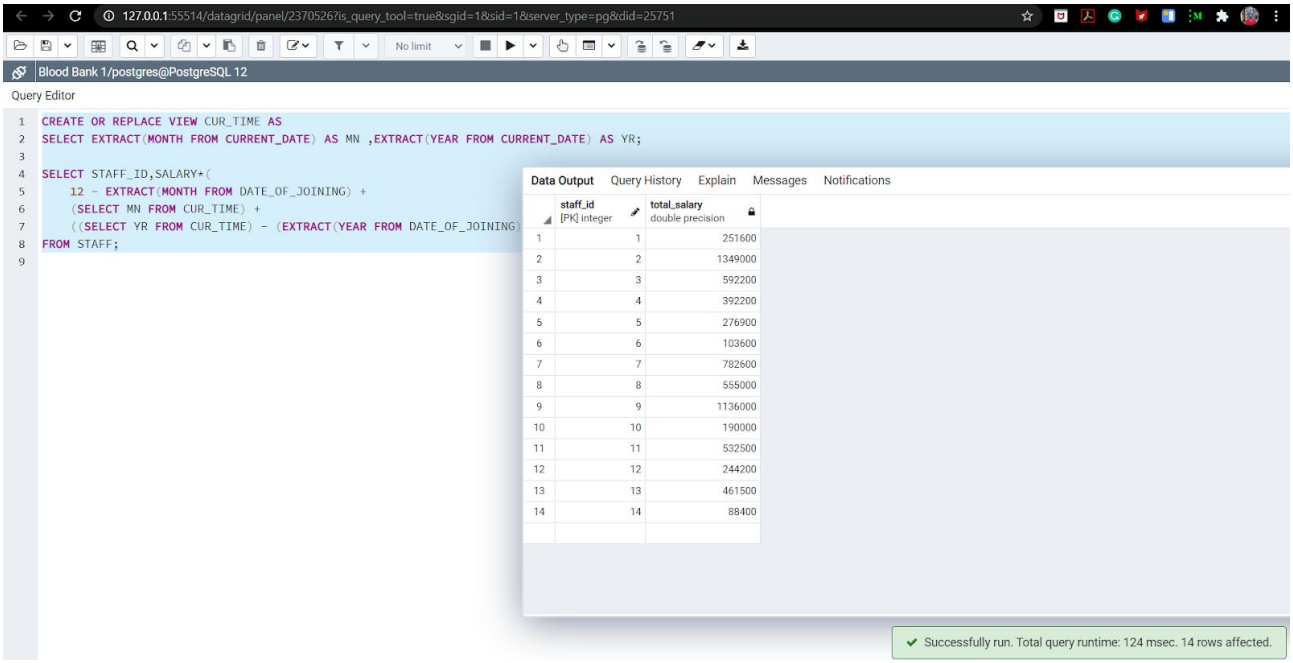
28. SELECT COUNTS.DT, SUM(COUNTS.SUM * ITEM.PRICE)
FROM (
SELECT EXTRACT(YEAR FROM DONATION_CAMP.EVENT_DATE) AS DT, ITEM_ID, SUM(INVENTORY.COUNT)
FROM DONATION_CAMP,INVENTORY
WHERE DONATION_CAMP.CAMP_ID = INVENTORY.CAMP_ID
GROUP BY ITEM_ID, EXTRACT(YEAR FROM EVENT_DATE)
ORDER BY ITEM_ID
) AS COUNTS NATURAL JOIN ITEM
GROUP BY COUNTS.DT
ORDER BY COUNTS.DT



Display Total Salary Received by each Staff Member till date

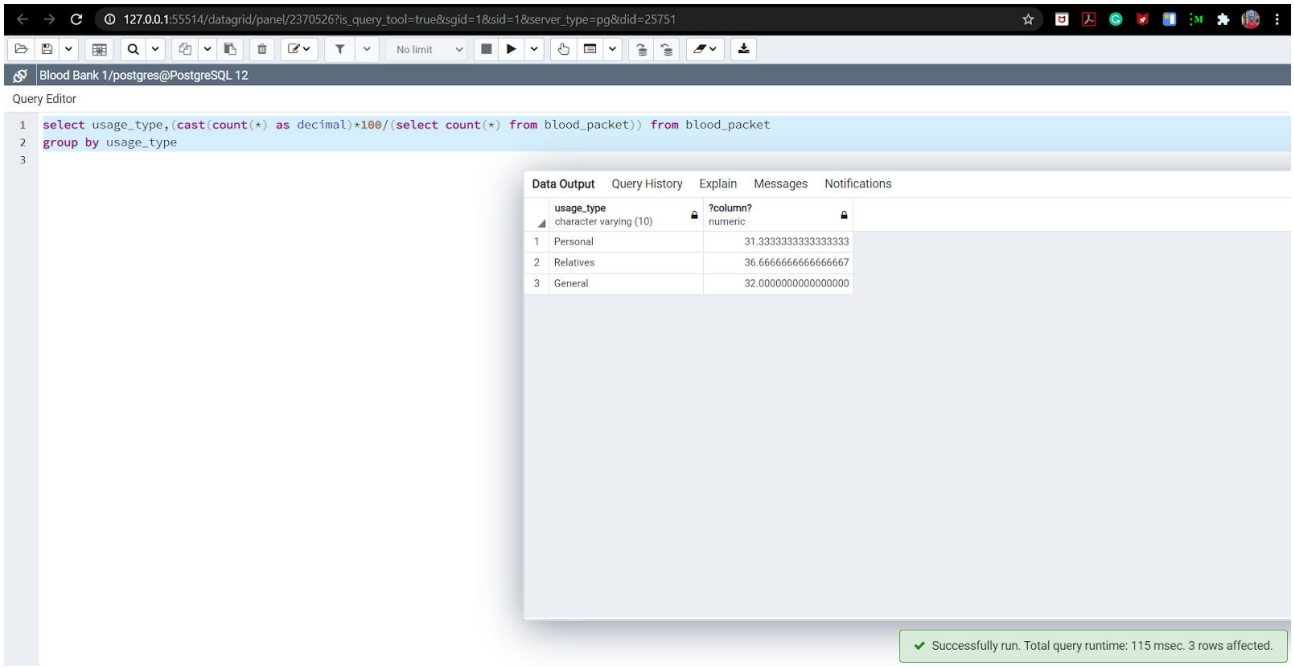
29. CREATE OR REPLACE VIEW CUR_TIME AS
SELECT EXTRACT(MONTH FROM CURRENT_DATE) AS MN ,EXTRACT(YEAR FROM CURRENT_DATE) AS YR

SELECT STAFF_ID,SALARY*(
12 - EXTRACT(MONTH FROM DATE_OF_JOINING) +
(SELECT MN FROM CUR_TIME) +
((SELECT YR FROM CUR_TIME) - (EXTRACT(YEAR FROM DATE_OF_JOINING)) - 1)*12) AS TOTAL_SALARY
FROM STAFF



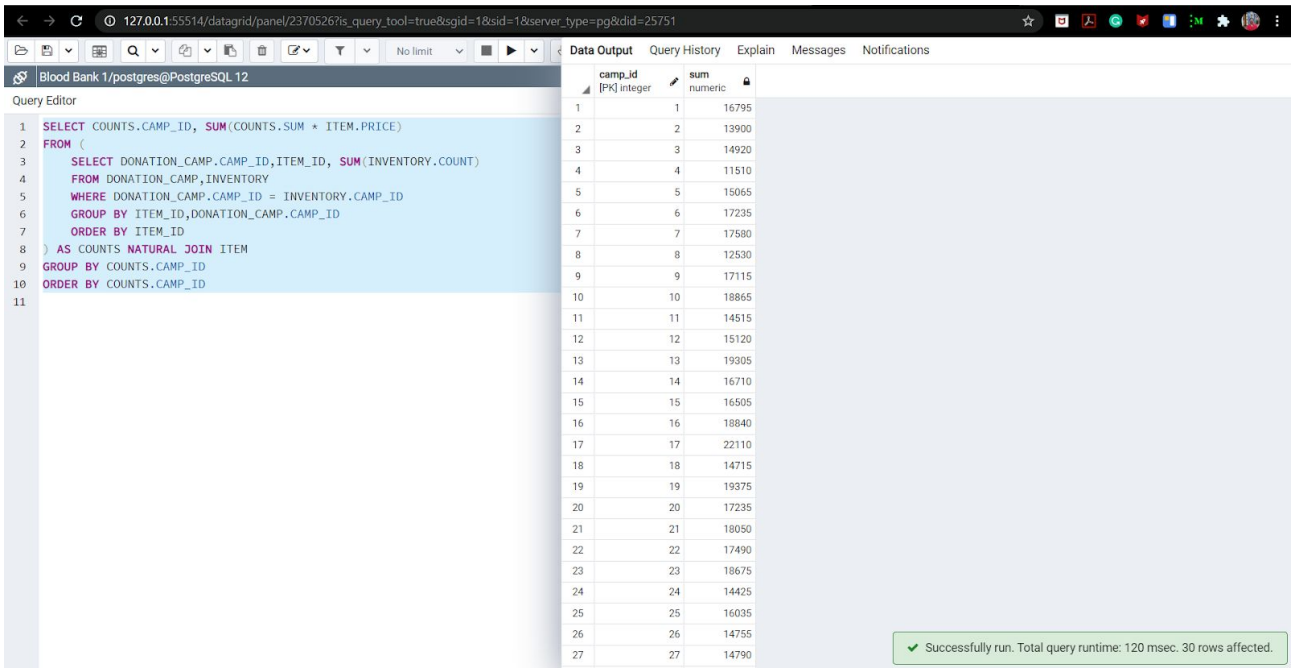
Display % of Packets for Each Category of USAGE_TYPE

30. select usage_type,(cast(count(*) as decimal)*100/(select count(*) from blood_packet)) from blood_packet
    group by usage_type



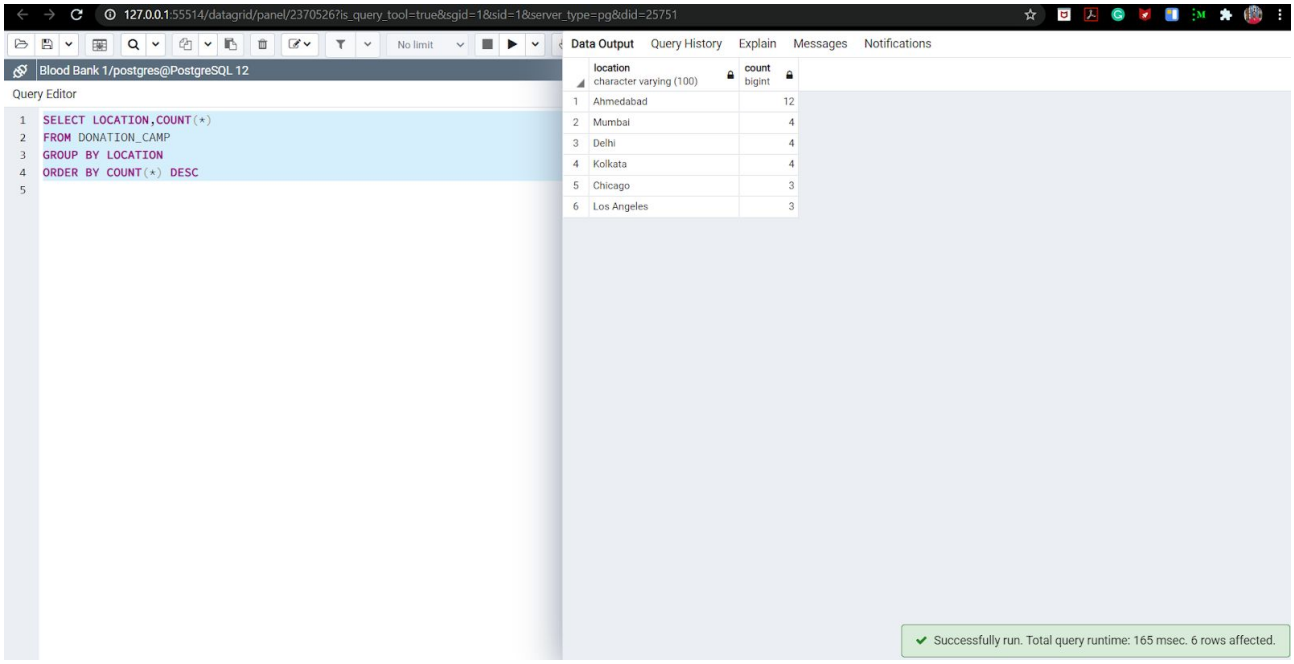
Display the Inventory Cost of Each Camp

31. SELECT COUNTS.CAMP_ID, SUM(COUNTS.SUM * ITEM.PRICE)
    FROM (
        SELECT DONATION_CAMP.CAMP_ID,ITEM_ID, SUM(INVENTORY.COUNT)
        FROM DONATION_CAMP,INVENTORY
        WHERE DONATION_CAMP.CAMP_ID = INVENTORY.CAMP_ID
        GROUP BY ITEM_ID,DONATION_CAMP.CAMP_ID
        ORDER BY ITEM_ID
) AS COUNTS NATURAL JOIN ITEM
    GROUP BY COUNTS.CAMP_ID
    ORDER BY COUNTS.CAMP_ID



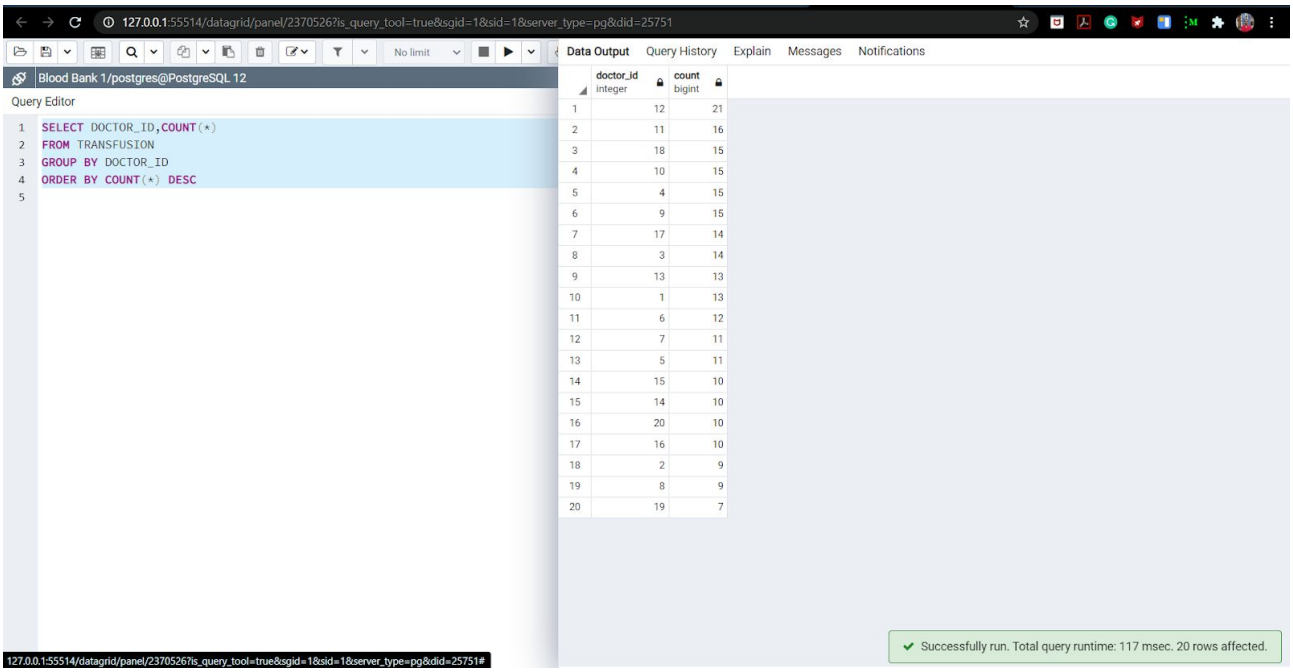
Rank the Cities Based on the Number of Camps Held

32. SELECT LOCATION,COUNT(*)
    FROM DONATION_CAMP
    GROUP BY LOCATION
    ORDER BY COUNT(*) DESC



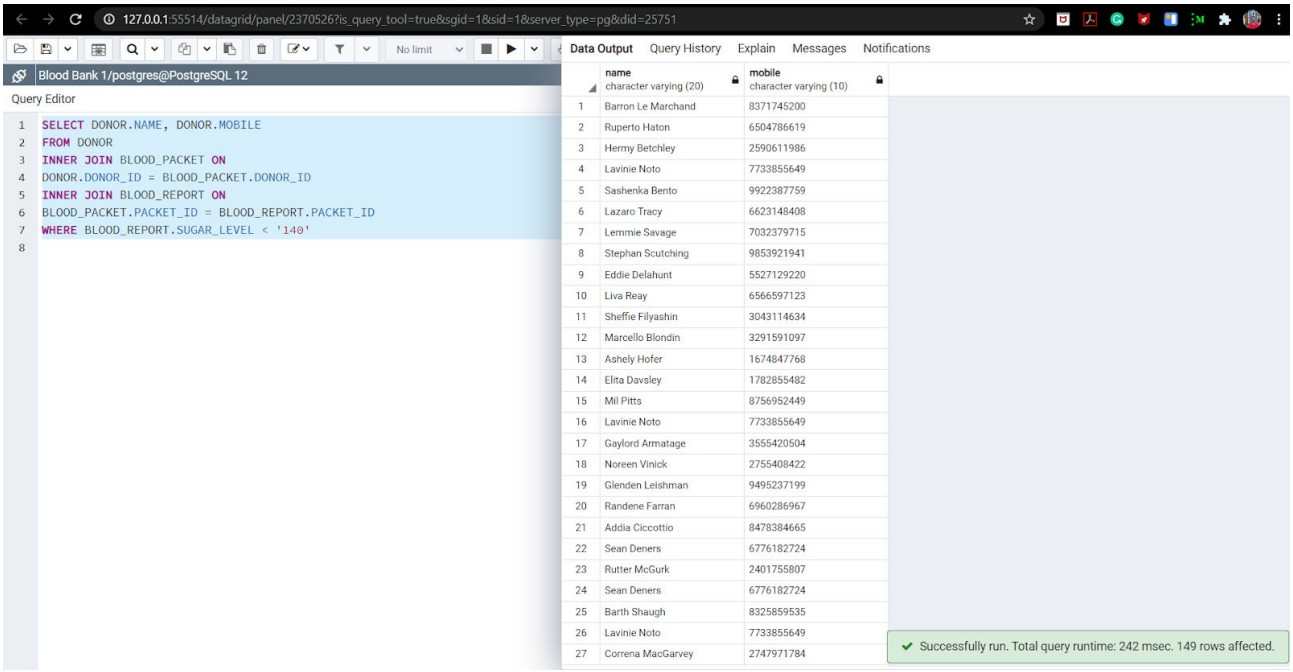
Rank the Doctors Based on the Number of Patients Sent

33. SELECT DOCTOR_ID,COUNT(*)
    FROM TRANSFUSION
    GROUP BY DOCTOR_ID
    ORDER BY COUNT(*) DESC



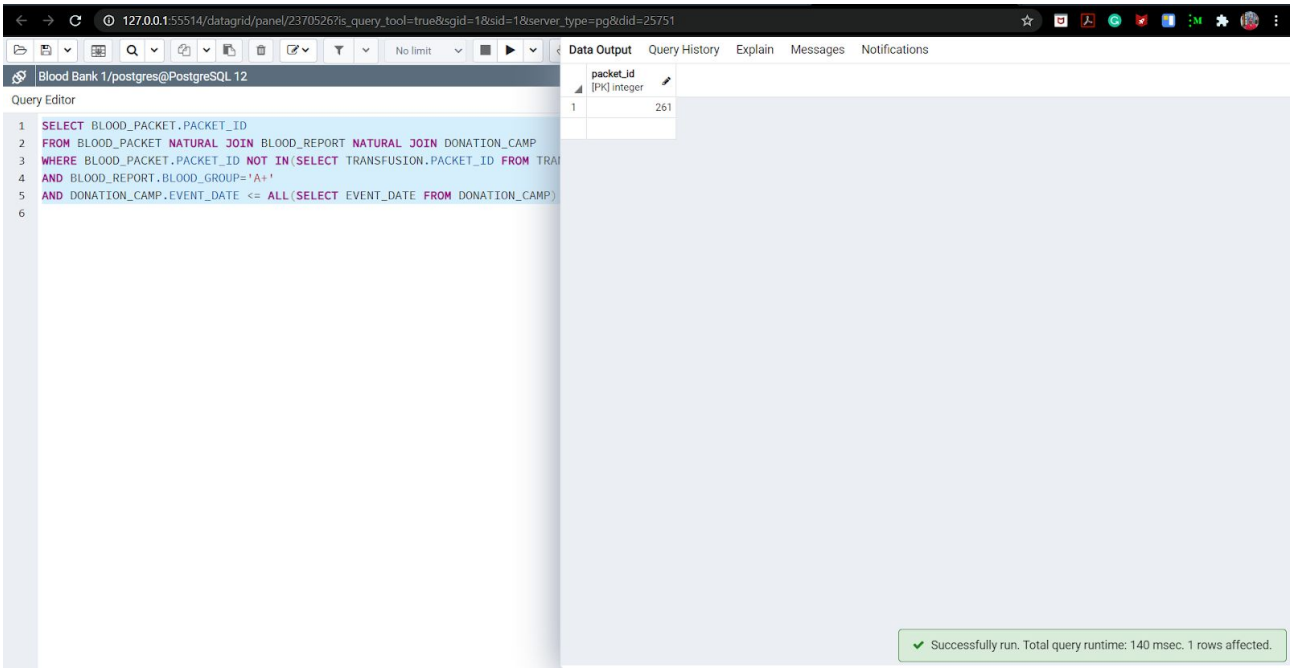
List down the Donors with a Normal Blood Sugar Level (below 140)

34. SELECT DONOR.NAME, DONOR.MOBILE
    FROM DONOR
    INNER JOIN BLOOD_PACKET ON
    DONOR.DONOR_ID = BLOOD_PACKET.DONOR_ID
    INNER JOIN BLOOD_REPORT ON
    BLOOD_PACKET.PACKET_ID = BLOOD_REPORT.PACKET_ID
    WHERE BLOOD_REPORT.SUGAR_LEVEL < '140'



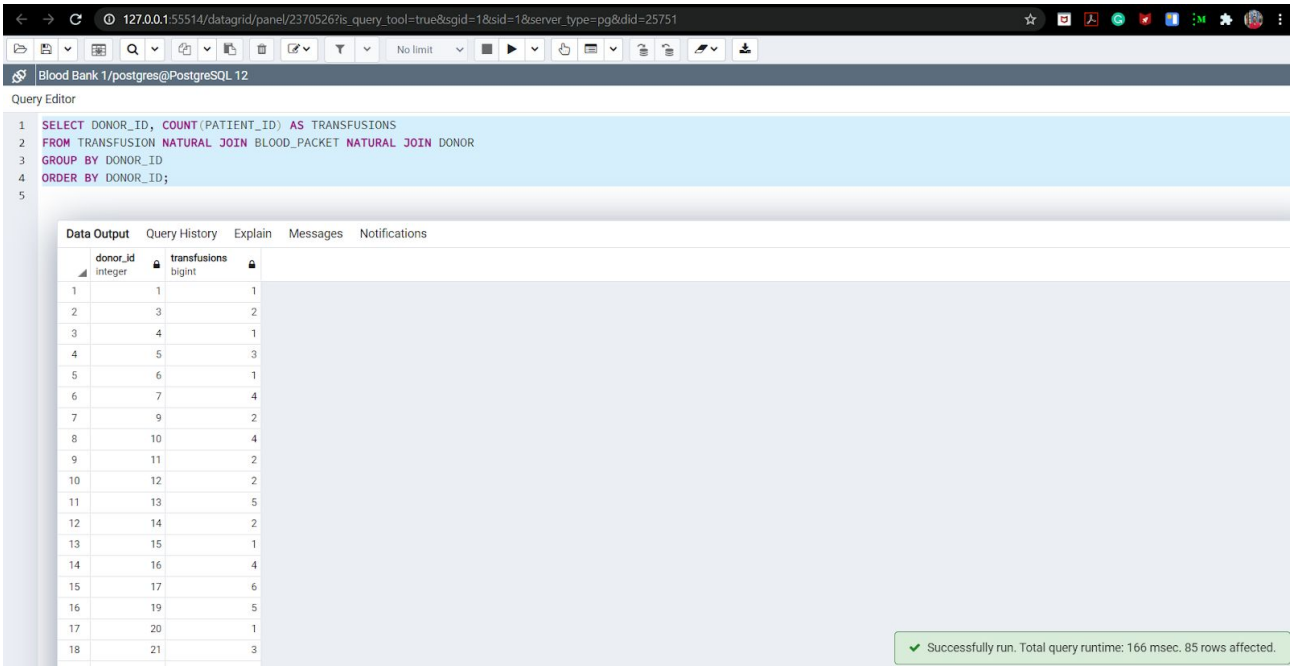
Return the Packet ID of the oldest unused blood packet with blood group A+

35. SELECT BLOOD_PACKET.PACKET_ID
    FROM BLOOD_PACKET NATURAL JOIN BLOOD_REPORT NATURAL JOIN DONATION_CAMP
    WHERE BLOOD_PACKET.PACKET_ID NOT IN(SELECT TRANSFUSION.PACKET_ID FROM TRANSFUSION)
    AND BLOOD_REPORT.BLOOD_GROUP='A+'
    AND DONATION_CAMP.EVENT_DATE <= ALL(SELECT EVENT_DATE FROM DONATION_CAMP);



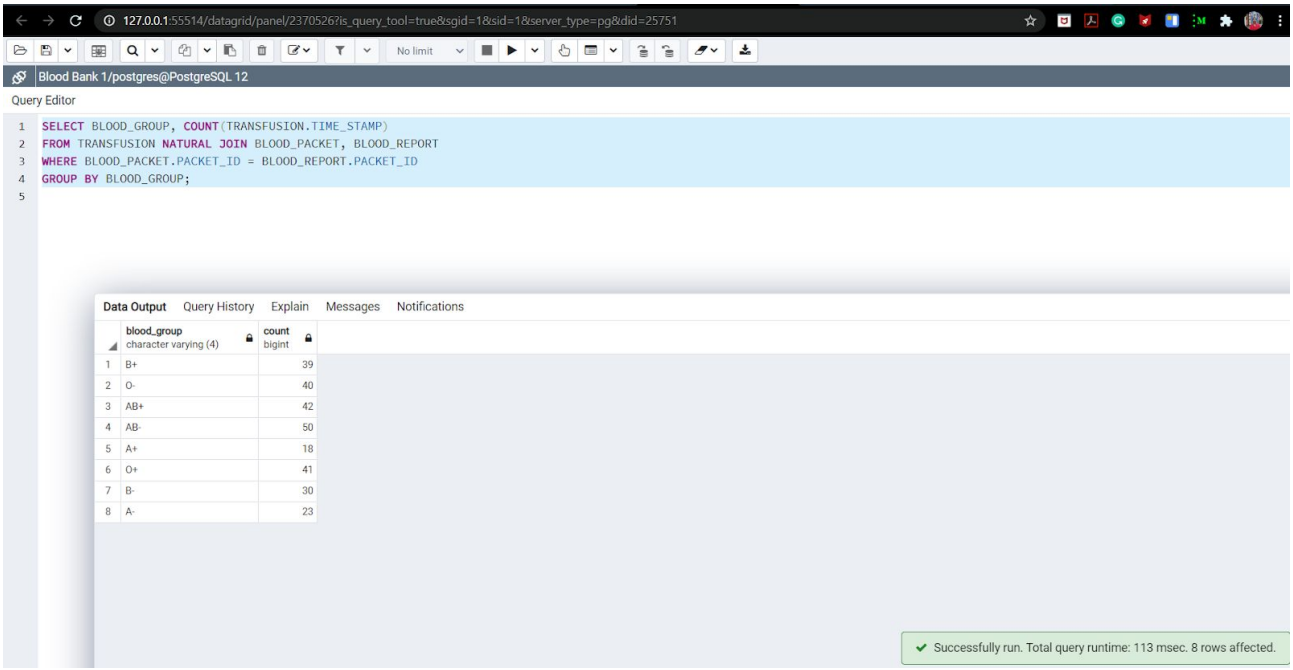
Number of Patients who received blood from a particular donor

36. SELECT DONOR_ID, COUNT(PATIENT_ID) AS TRANSFUSIONS
    FROM TRANSFUSION NATURAL JOIN BLOOD_PACKET NATURAL JOIN DONOR
    GROUP BY DONOR_ID
    ORDER BY DONOR_ID;



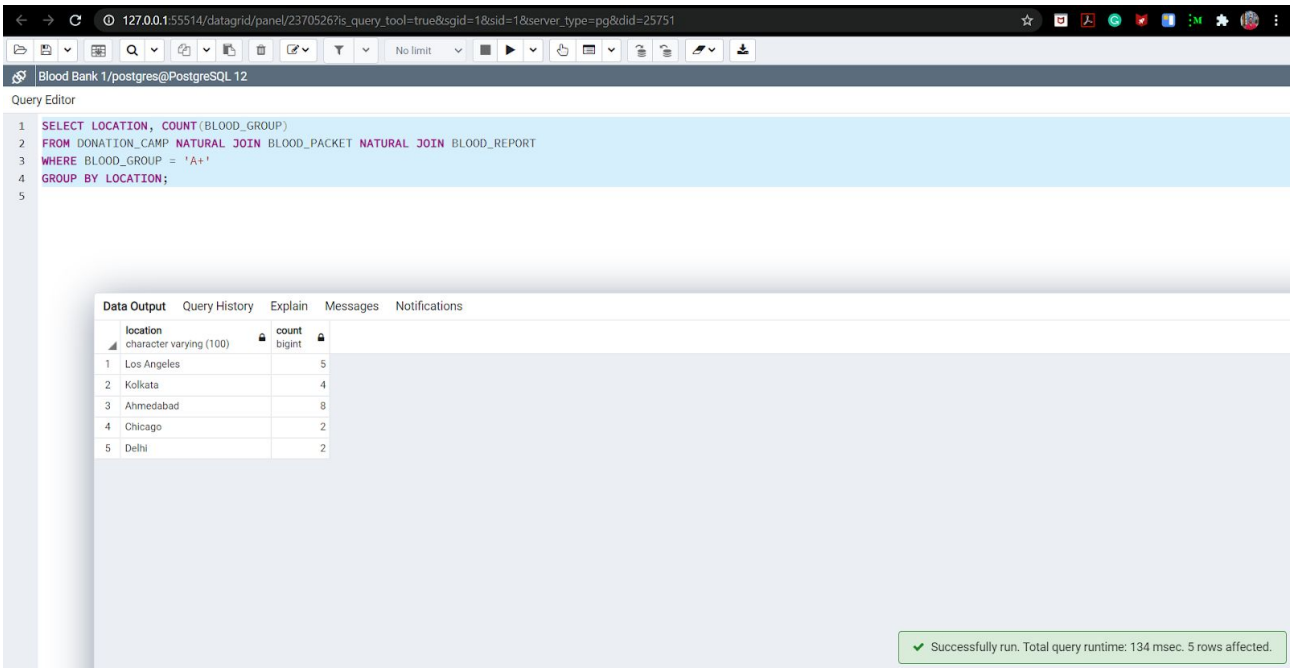
Number of blood packets of each blood group used in transfusion

37. SELECT BLOOD_GROUP, COUNT(TRANSFUSION.TIME_STAMP)
    FROM TRANSFUSION NATURAL JOIN BLOOD_PACKET, BLOOD_REPORT
    WHERE BLOOD_PACKET.PACKET_ID = BLOOD_REPORT.PACKET_ID
    GROUP BY BLOOD_GROUP;



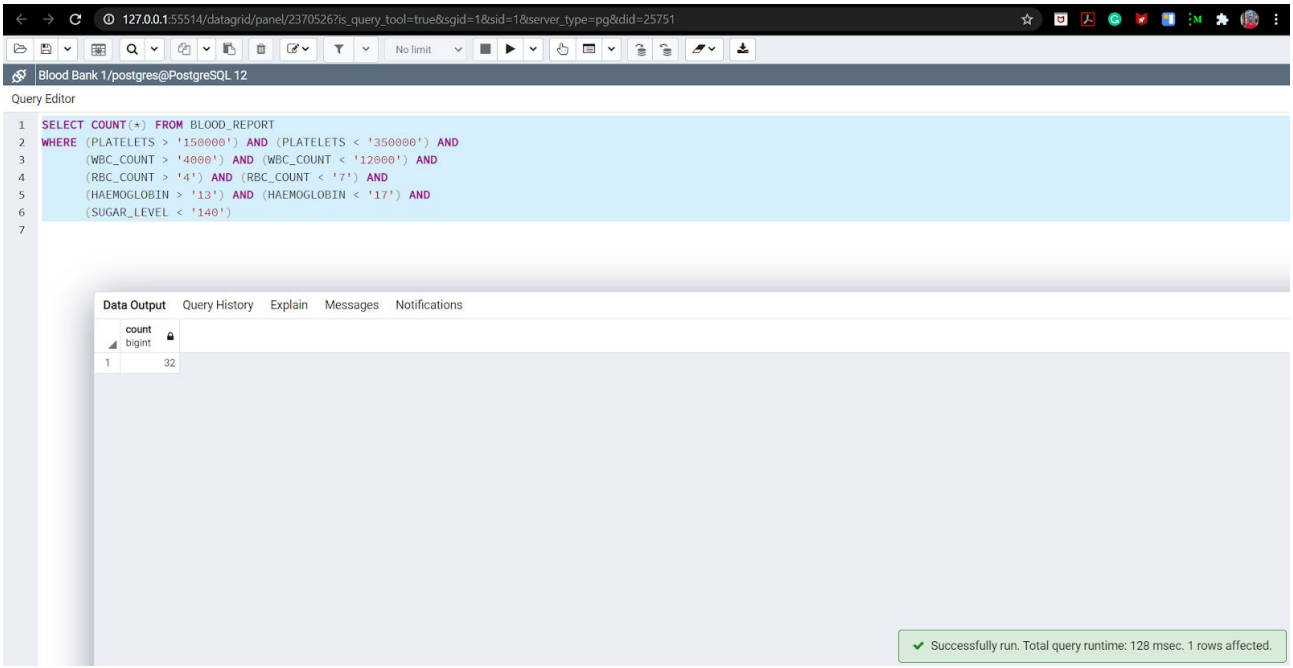
Number of Donations with Blood Group A+ from each location

38. **SELECT** LOCATION, **COUNT(**BLOOD_GROUP**)**
    FROM DONATION_CAMP **NATURAL JOIN** BLOOD_PACKET **NATURAL JOIN** BLOOD_REPORT
    WHERE BLOOD_GROUP = 'A+'
    GROUP BY LOCATION;



Number of healthy donors i.e. platelet count between 150000 and 350000, WBC between 4000 and 12000, RBC between 4 and 7, Haemoglobin between 13 and 17, Sugar level less than 140.

39. **SELECT COUNT(*) FROM** BLOOD_REPORT
    WHERE (PLATELETS > '150000') **AND** (PLATELETS < '350000') **AND**
    (WBC_COUNT > '4000') **AND** (WBC_COUNT < '12000') **AND**
    (RBC_COUNT > '4') **AND** (RBC_COUNT < '7') **AND**
    (HAEMOGLOBIN > '13') **AND** (HAEMOGLOBIN < '17') **AND**
    (SUGAR_LEVEL < '140')



List of Patient and Donor of Each Transfusion

40. **SELECT** DONOR.NAME **AS** DONOR, PATIENT.NAME **AS** PATIENT
    FROM DONOR **NATURAL JOIN** BLOOD_PACKET, TRANSFUSION, PATIENT
    WHERE BLOOD_PACKET.PACKET_ID = TRANSFUSION.PACKET_ID **AND**
    TRANSFUSION.PATIENT_ID = PATIENT.PATIENT_ID;