Ahmed Muntasir Hossain

CSCI 2215

5a – Business Section

a - Explain what the business is that hired you to develop a database

Description of the Business: Connecticut Blood Bank is a blood collection center (organization). The donor provides blood, which is collected, typed, separated into components, stored, and prepared for transfusion to recipients. The bank has multiple branches where donors visit to donate blood. The blood is stored in bags and patients purchase those blood bags. The blood is processed in the laboratories in each branch. The branches have equipment for collection, separation, storage, and donation. Each branch has employees and managers who work there and oversee the daily operations.

b - Provide a brief description of the problem that the business has

The blood bank was created in 1999. It was initially based in New Haven for the past 10 years, however with greater demands from neighboring cities, the organization had to expand to other cities. The bank used an Excel Sheet to keep track of the donors, patients, and the few branches it had recently expanded to. However, there was a recent case where a patient received a contaminated blood bag and the company decided to change their organizational scheme. They decided to develop a database to store detailed information about the entire process. The process is detailed below in the business constraints. Primarily, the company would need to be able to track the blood bag being donated by a donor. The status of the blood would be recorded as it passes through the laboratory and finally to the patient. Medical information of the donor and patients would also need to be on file. Additionally, they decided to put branches in 10 other major cities in CT and realized the importance of keeping track of all the data, employees, inventory, and equipment, in all their branches. The primarily problem is that the database for the company has to be built from the beginning compared to other organizations where the databases are usually updated over time.

c - Provide a series of specifications that the business needs

1. The company is organized into **branches.**
2. Each branch has a location (**street address, city, state, country, zip code**) and an **ID.**
3. Each branch may have a **manager** who is identified by her/his **SSN number.**

4. All **employees** are identified by their **SSN numbers, first name, and last name**.
5. The **manager** is an **employee** of the company.
6. An **employee** may have a **manager** under whom the **employee** works.
7. An **employee** works in one particular **branch**, and the **manager** of that **branch** (if exists) would be the **manager** of that **employee**.

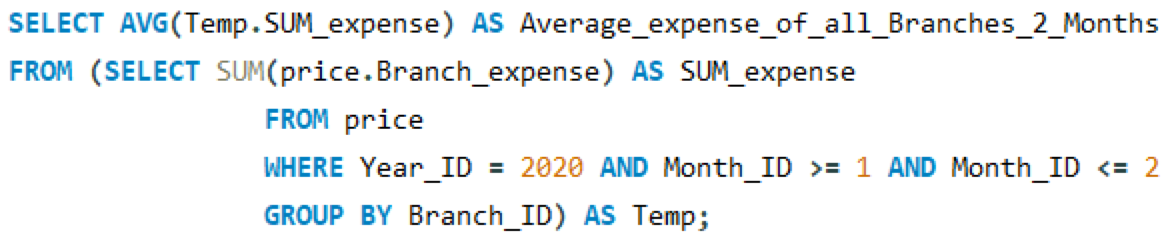
8. A **donor** provides their **first name, last name, blood group and any underlying condition**. A donor is identified by their **ID** number. The underlying conditions are specific diseases that the donor is tested for.
9. A **patient** provides their **first name, last name, blood group.** A patient is identified by their **ID** number. The underlying condition of a patient is not required since the patient is receiving the transfusion.

10. A **donor** donates a bag of blood, which is identified by its **blood bag ID**. The particular **branch** and the **date** on which the blood is donated is recorded. Additionally, type of donation: (**Red blood cells, platelets, plasma**) is recorded.
11. A **patient** goes to a particular **branch** for a **donor blood bag** on a particular **date**.
12. We keep track of which blood bag (**blood bag ID)** is given to which patient (**patient ID)** by using a separate **transfusion ID.**
13. We also keep track of the **status** of a blood bag (**In Storage, Laboratory, Out of Storage**). The moment the **status** of the blood bag changes, we record the time (**stamp).**

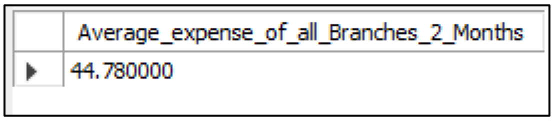
14. Each **branch** has a **monthly expense** which is recorded. The unit price of **RBC, platelets, and plasma** may change so the unit price is stored for each branch.
15. Quantities of **lab equipments, incubators, storage equipments, and donation equipments** for each **branch** is recorded.

d - Provide at least 8 reports that the business would want to see

SUB QUERY 1: Returns the average expense of all the branches from January 2020 to February 2020.

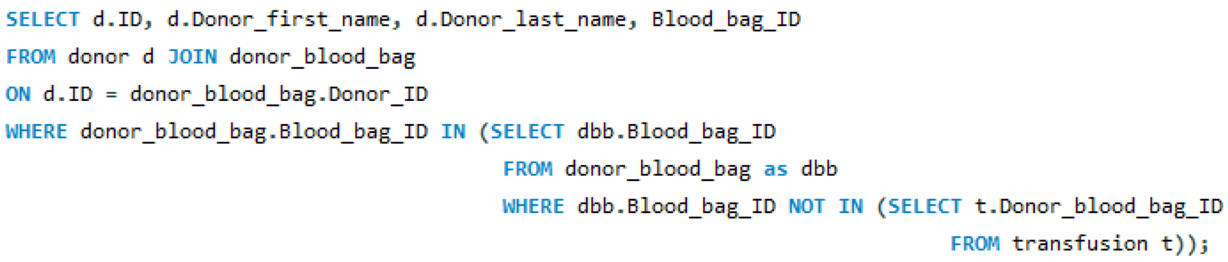


```
SELECT AVG(Temp.SUM_expense) AS Average_expense_of_all_Branches_2_Months
FROM (SELECT SUM(price.Branch_expense) AS SUM_expense
                FROM price
                WHERE Year_ID = 2020 AND Month_ID >= 1 AND Month_ID <= 2
                GROUP BY Branch_ID) AS Temp;
```

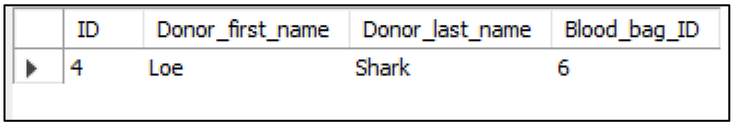


	Average_expense_of_all_Branches_2_Months
▶	44.780000

SUB QUERY 2: Returns the ID, first name, and last name of the donor who donated a blood bag that has not been transferred to a patient yet. The blood bag ID is also returned.

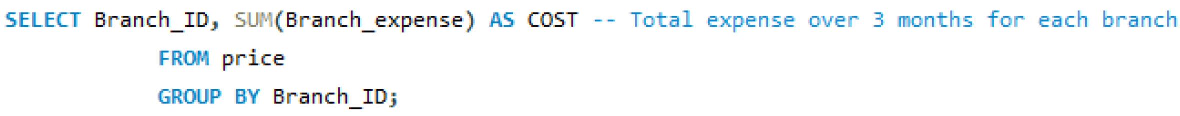


```
SELECT d.ID, d.Donor_first_name, d.Donor_last_name, Blood_bag_ID
FROM donor d JOIN donor_blood_bag
ON d.ID = donor_blood_bag.Donor_ID
WHERE donor_blood_bag.Blood_bag_ID IN (SELECT dbb.Blood_bag_ID
                                        FROM donor_blood_bag as dbb
                                        WHERE dbb.Blood_bag_ID NOT IN (SELECT t.Donor_blood_bag_ID
                                                                        FROM transfusion t));
```

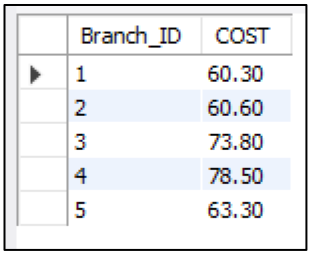


	ID	Donor_first_name	Donor_last_name	Blood_bag_ID
▶	4	Loe	Shark	6

REGULAR QUERY 1: Returns the total expense for each branch from the starting date of that branch. The branch IDs of those branches are also returned.



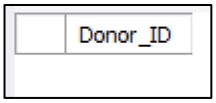
```
SELECT Branch_ID, SUM(Branch_expense) AS COST -- Total expense over 3 months for each branch
            FROM price
            GROUP BY Branch_ID;
```



Branch_ID	COST
1	60.30
2	60.60
3	73.80
4	78.50
5	63.30

REGULAR QUERY 2: Returns the ID of the donor who has tested positive for any of the underlying conditions.

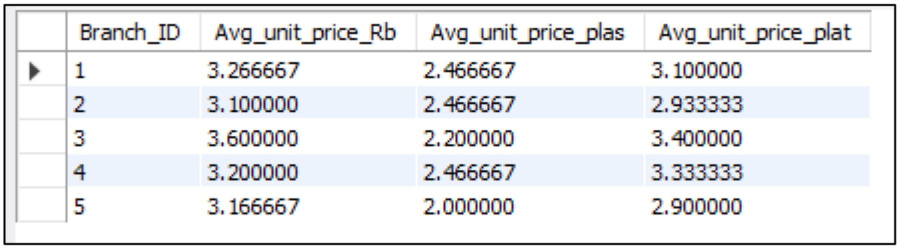
```
SELECT Donor_ID
FROM donor_underlying_condition
    WHERE HIV_1 = 1
    OR HTLV_I = 1
    OR HTLV_II = 1
    OR Hepatitis_C = 1
    OR Hepatitis_B = 1
    OR West_Nile_Virus = 1
    OR T_pallidum= 1;
```



Donor_ID

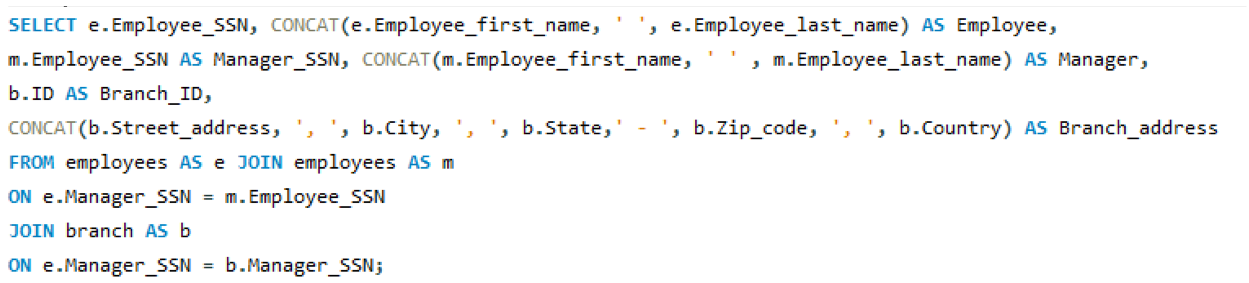
REGULAR QUERY 3: Returns the average unit price of each component of blood, for each branch, from the starting date of that branch. The corresponding branch ID is also returned.

```
SELECT Branch_ID, AVG(Unit_price_rb) AS Avg_unit_price_Rb,
AVG(Unit_price_plas) AS Avg_unit_price_plas, AVG(Unit_price_plat) AS Avg_unit_price_plat
    FROM price
    GROUP BY Branch_ID;
```

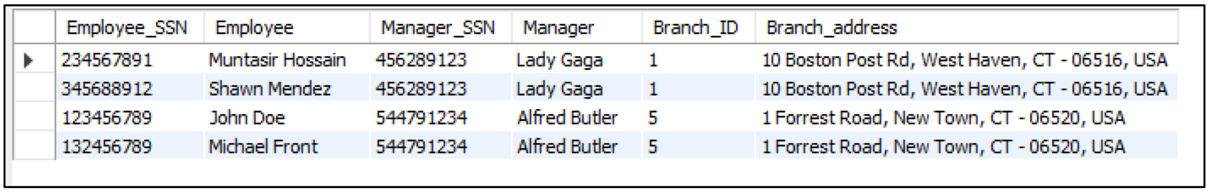


Branch_ID	Avg_unit_price_Rb	Avg_unit_price_plas	Avg_unit_price_plat
1	3.266667	2.466667	3.100000
2	3.100000	2.466667	2.933333
3	3.600000	2.200000	3.400000
4	3.200000	2.466667	3.333333
5	3.166667	2.000000	2.900000

JOIN 1: Returns the SSN and name of only those employees who have a manager and who work under a particular branch. The corresponding manager SSN, manager name, branch ID and branch address are also returned.

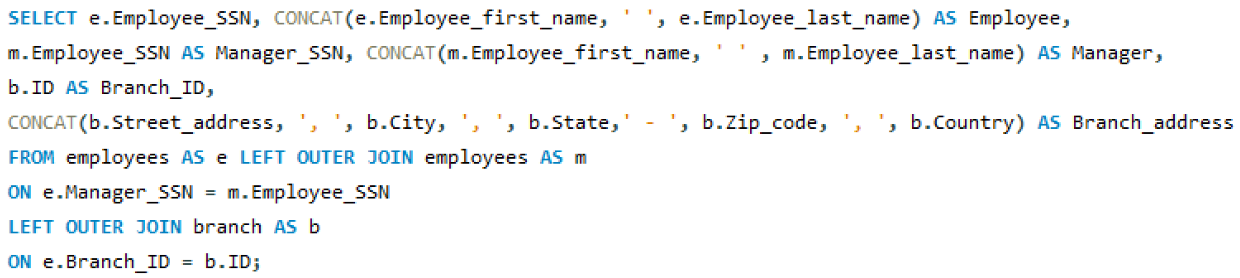


```
SELECT e.Employee_SSN, CONCAT(e.Employee_first_name, ' ', e.Employee_last_name) AS Employee,
m.Employee_SSN AS Manager_SSN, CONCAT(m.Employee_first_name, ' ' , m.Employee_last_name) AS Manager,
b.ID AS Branch_ID,
CONCAT(b.Street_address, ', ', b.City, ', ', b.State,' - ', b.Zip_code, ', ', b.Country) AS Branch_address
FROM employees AS e JOIN employees AS m
ON e.Manager_SSN = m.Employee_SSN
JOIN branch AS b
ON e.Manager_SSN = b.Manager_SSN;
```

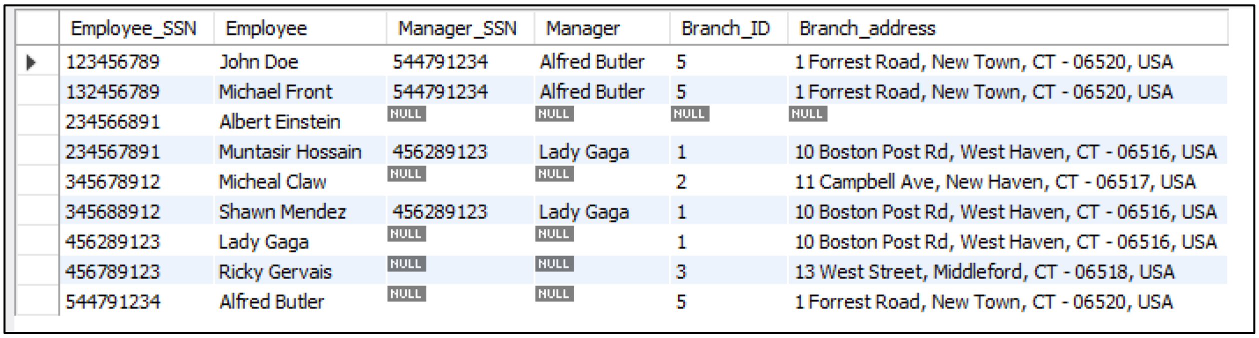


Employee_SSN	Employee	Manager_SSN	Manager	Branch_ID	Branch_address
234567891	Muntasir Hossain	456289123	Lady Gaga	1	10 Boston Post Rd, West Haven, CT - 06516, USA
345688912	Shawn Mendez	456289123	Lady Gaga	1	10 Boston Post Rd, West Haven, CT - 06516, USA
123456789	John Doe	544791234	Alfred Butler	5	1 Forrest Road, New Town, CT - 06520, USA
132456789	Michael Front	544791234	Alfred Butler	5	1 Forrest Road, New Town, CT - 06520, USA

JOIN 2: Returns the SSN and name of ALL the employees in the organization, their respective manager (if any) and branch (if any) under which they work.



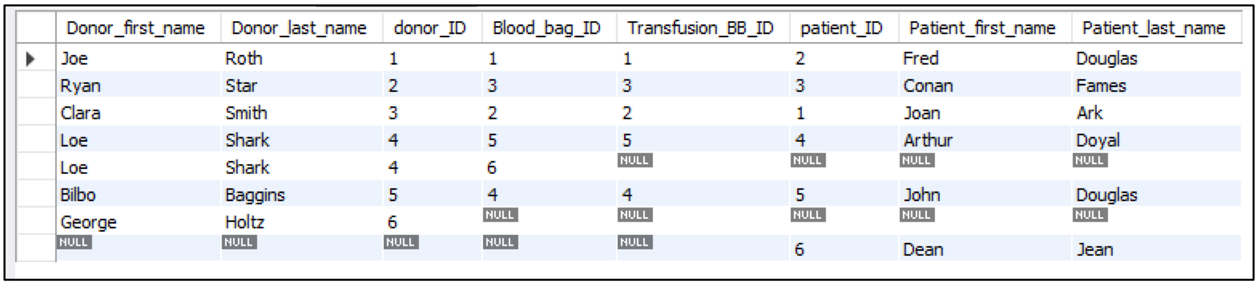
```
SELECT e.Employee_SSN, CONCAT(e.Employee_first_name, ' ', e.Employee_last_name) AS Employee,
m.Employee_SSN AS Manager_SSN, CONCAT(m.Employee_first_name, ' ' , m.Employee_last_name) AS Manager,
b.ID AS Branch_ID,
CONCAT(b.Street_address, ', ', b.City, ', ', b.State,' - ', b.Zip_code, ', ', b.Country) AS Branch_address
FROM employees AS e LEFT OUTER JOIN employees AS m
ON e.Manager_SSN = m.Employee_SSN
LEFT OUTER JOIN branch AS b
ON e.Branch_ID = b.ID;
```



Employee_SSN	Employee	Manager_SSN	Manager	Branch_ID	Branch_address
123456789	John Doe	544791234	Alfred Butler	5	1 Forrest Road, New Town, CT - 06520, USA
132456789	Michael Front	544791234	Alfred Butler	5	1 Forrest Road, New Town, CT - 06520, USA
234566891	Albert Einstein	NULL	NULL	NULL	NULL
234567891	Muntasir Hossain	456289123	Lady Gaga	1	10 Boston Post Rd, West Haven, CT - 06516, USA
345678912	Micheal Claw	NULL	NULL	2	11 Campbell Ave, New Haven, CT - 06517, USA
345688912	Shawn Mendez	456289123	Lady Gaga	1	10 Boston Post Rd, West Haven, CT - 06516, USA
456289123	Lady Gaga	NULL	NULL	1	10 Boston Post Rd, West Haven, CT - 06516, USA
456789123	Ricky Gervais	NULL	NULL	3	13 West Street, Middleford, CT - 06518, USA
544791234	Alfred Butler	NULL	NULL	5	1 Forrest Road, New Town, CT - 06520, USA

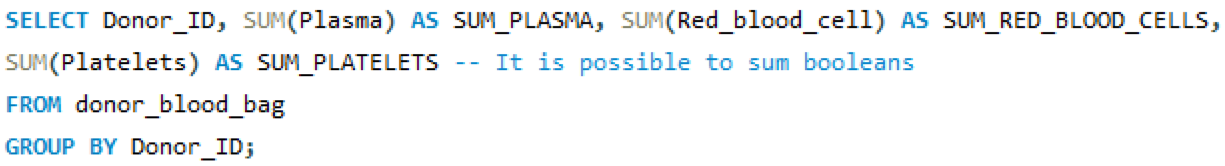
JOIN 3: Returns the name and ID of ALL the donors and their respective patients (if any) and ALL the patients and their respective donors (if any) in the database. Additionally, the ID of the blood bag that has been donated by a donor (Blood_bag_ID), and the ID of the blood bag that has been transferred to a patient (Transfusion_BB_ID) are also returned. For instance, a blood bag may have been donated that is currently being processed. Therefore, the Blood_bag_ID would have a value (ID) but the Transfusion_BB_ID would be null.

```sql
CREATE VIEW dbb AS
SELECT d.ID, d.Donor_first_name, d.Donor_last_name, db.Blood_bag_ID
FROM donor AS d LEFT OUTER JOIN donor_blood_bag as db
ON  d.ID = db.Donor_ID;

SELECT * FROM dbb;

SELECT dbb.Donor_first_name, dbb.Donor_last_name, dbb.ID AS donor_ID,
dbb.Blood_bag_ID, t.Donor_blood_bag_ID AS Transfusion_BB_ID,
p.ID AS patient_ID, p.Patient_first_name, p.Patient_last_name
FROM dbb LEFT OUTER JOIN transfusion as t
ON dbb.Blood_bag_ID = t.Donor_blood_bag_ID
LEFT OUTER JOIN patient as p
ON t.Patient_ID = p.ID
UNION
SELECT dbb.Donor_first_name, dbb.Donor_last_name, dbb.ID AS donor_ID, dbb.Blood_bag_ID,
t.Donor_blood_bag_ID AS Transfusion_BB_ID, p.ID AS patient_ID, p.Patient_first_name, p.Patient_last_name
FROM dbb RIGHT OUTER JOIN transfusion as t
ON dbb.Blood_bag_ID = t.Donor_blood_bag_ID
RIGHT OUTER JOIN patient as p
ON t.Patient_ID = p.ID;
```

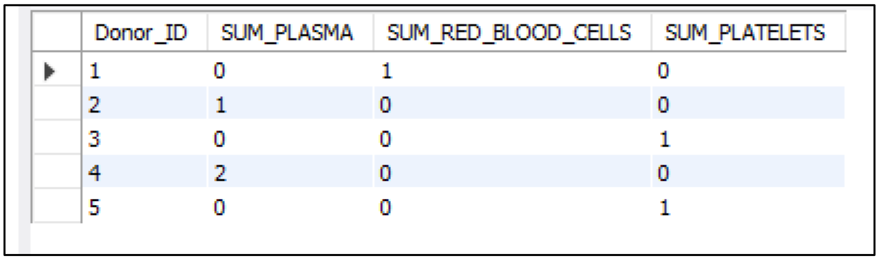


Donor_first_name	Donor_last_name	donor_ID	Blood_bag_ID	Transfusion_BB_ID	patient_ID	Patient_first_name	Patient_last_name
Joe	Roth	1	1	1	2	Fred	Douglas
Ryan	Star	2	3	3	3	Conan	Fames
Clara	Smith	3	2	2	1	Joan	Ark
Loe	Shark	4	5	5	4	Arthur	Doyal
Loe	Shark	4	6	NULL	NULL	NULL	NULL
Bilbo	Baggins	5	4	4	5	John	Douglas
George	Holtz	6	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	6	Dean	Jean

GROUP BY 1: Returns the total number of blood bags donated by a particular donor, for each component of blood.

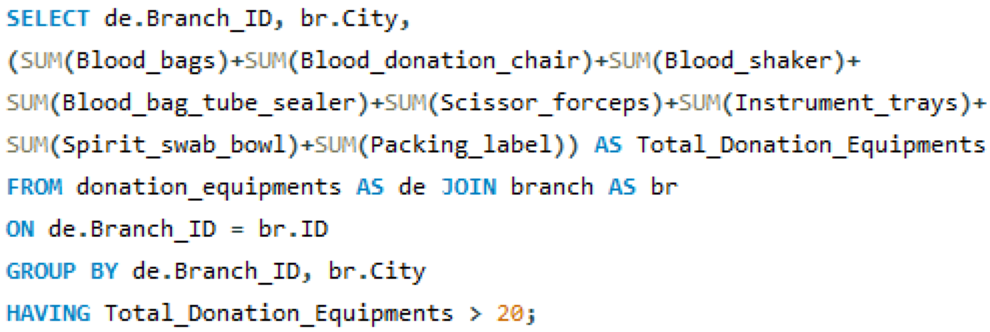


```
SELECT Donor_ID, SUM(Plasma) AS SUM_PLASMA, SUM(Red_blood_cell) AS SUM_RED_BLOOD_CELLS,
SUM(Platelets) AS SUM_PLATELETS -- It is possible to sum booleans
FROM donor_blood_bag
GROUP BY Donor_ID;
```

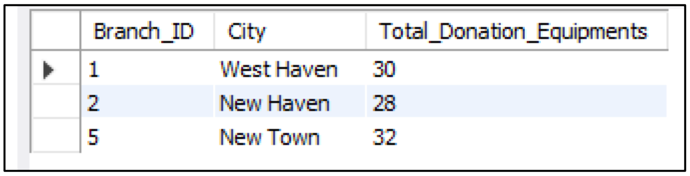


Donor_ID	SUM_PLASMA	SUM_RED_BLOOD_CELLS	SUM_PLATELETS
1	0	1	0
2	1	0	0
3	0	0	1
4	2	0	0
5	0	0	1

GROUP BY 2: Returns the ID and city of the branch where the total number of donation equipment is greater than 20. The corresponding number of donation equipment is also returned.

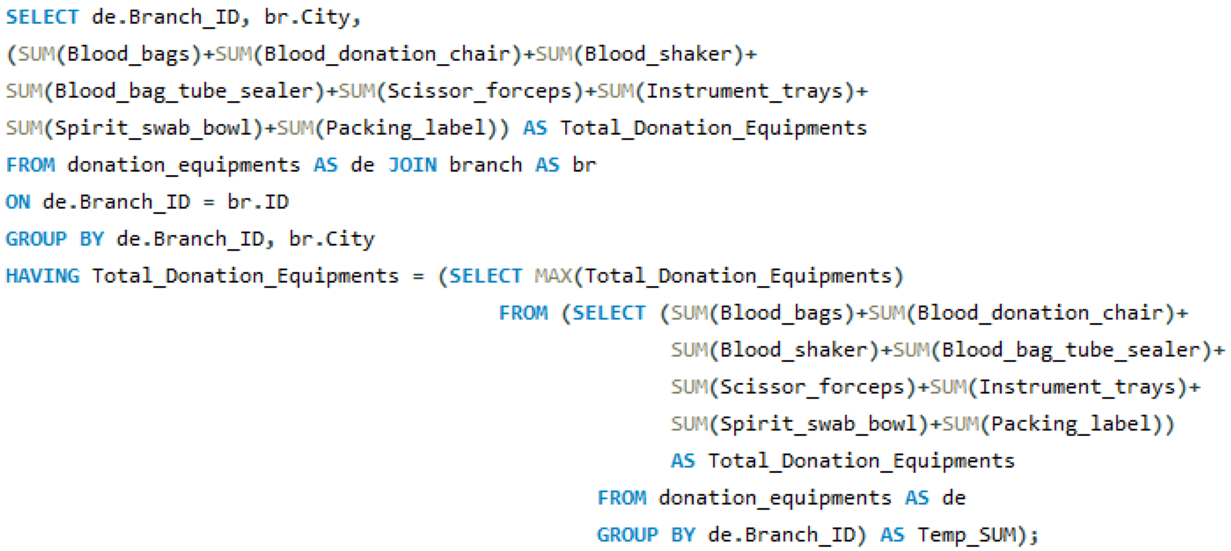


```
SELECT de.Branch_ID, br.City,
(SUM(Blood_bags)+SUM(Blood_donation_chair)+SUM(Blood_shaker)+
SUM(Blood_bag_tube_sealer)+SUM(Scissor_forceps)+SUM(Instrument_trays)+
SUM(Spirit_swab_bowl)+SUM(Packing_label)) AS Total_Donation_Equipments
FROM donation_equipments AS de JOIN branch AS br
ON de.Branch_ID = br.ID
GROUP BY de.Branch_ID, br.City
HAVING Total_Donation_Equipments > 20;
```

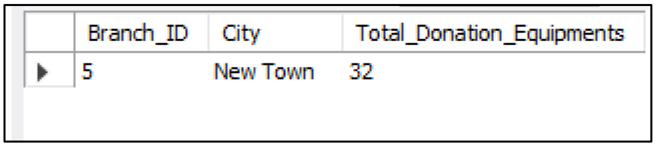


Branch_ID	City	Total_Donation_Equipments
1	West Haven	30
2	New Haven	28
5	New Town	32

GROUP BY 3: Returns the ID and city of the branch with the maximum number of donation equipment. The corresponding number of donation equipment is also returned.



```
SELECT de.Branch_ID, br.City,
(SUM(Blood_bags)+SUM(Blood_donation_chair)+SUM(Blood_shaker)+
SUM(Blood_bag_tube_sealer)+SUM(Scissor_forceps)+SUM(Instrument_trays)+
SUM(Spirit_swab_bowl)+SUM(Packing_label)) AS Total_Donation_Equipments
FROM donation_equipments AS de JOIN branch AS br
ON de.Branch_ID = br.ID
GROUP BY de.Branch_ID, br.City
HAVING Total_Donation_Equipments = (SELECT MAX(Total_Donation_Equipments)
                                        FROM (SELECT (SUM(Blood_bags)+SUM(Blood_donation_chair)+
                                                      SUM(Blood_shaker)+SUM(Blood_bag_tube_sealer)+
                                                      SUM(Scissor_forceps)+SUM(Instrument_trays)+
                                                      SUM(Spirit_swab_bowl)+SUM(Packing_label))
                                                      AS Total_Donation_Equipments
                                                FROM donation_equipments AS de
                                                GROUP BY de.Branch_ID) AS Temp_SUM);
```



Branch_ID	City	Total_Donation_Equipments
5	New Town	32