# Output File: Group 10
# BLOOD DONATION ORGANIZATION
# BUAN 6320

Suma Nednurkar

## Simple Queries

1. SELECT COUNT(*) AS 'Number of Staff' from staff_t;

```
820
821   ###### SIMPLE QUERIES ######
822   /* This query will return the number of staff in the "staff_T" table */
823 • SELECT COUNT(*) AS 'Number of Staff' from staff_t;
824
825   /*This query will return the information on donors who have not made any blood donations. */
826 • SELECT * FROM donor_T WHERE last_donation IS NULL;
827
828   /* This query displays the full name of donors in an easy-to-read format.
829   Using the CONCAT function, the first and last name are returned in a single column and we have us
830   to label that returning column. */
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| Number of Staff |
|---|
| 50 |

2. SELECT * FROM donor_T WHERE last_donation IS NULL;

```
821   ###### SIMPLE QUERIES ######
822   /* This query will return the number of staff in the "staff_T" table */
823 • SELECT COUNT(*) AS 'Number of Staff' from staff_t;
824
825   /*This query will return the information on donors who have not made any blood donations. */
826 • SELECT * FROM donor_T WHERE last_donation IS NULL;
827
828   /* This query displays the full name of donors in an easy-to-read format.
829   Using the CONCAT function, the first and last name are returned in a single column and we have use
830   to label that returning column. */
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| donor_id | first_name | last_name | email | phone | blood_type | gender | birth_date | weight | last_donation |
|---|---|---|---|---|---|---|---|---|---|
| 100 | Johnny | Doe | john.doe@example.com | 1234567890 | A+ | Male | 1990-01-01 | 150 | NULL |
| 102 | Bob | Smith | bob.smith@example.com | 1112223333 | AB- | Male | 1985-03-03 | 170 | NULL |
| 106 | Tom | Wilson | tom.wilson@example.com | 5556667777 | B- | Male | 1996-07-07 | 155 | NULL |
| 109 | Amy | White | amy.white@example.com | 2223334444 | A- | Female | 1993-10-10 | 125 | NULL |
| 112 | Dave | Carter | dave.carter@example.com | 1112223333 | O+ | Male | 1997-01-13 | 175 | NULL |

donor_T 55 ×

Output

3. SELECT CONCAT(first_name, ' ', last_name) AS `Donor Full Name` FROM donor_T;

```
577 •  SELECT CONCAT(first_name, ' ', last_name) AS `Donor Full Name` FROM donor_T;
578
579
580
581
```

100% 77:577

Result Grid Filter Rows: Search Export:

| Donor Full Name |
|---|
| Johnny Doe |
| Jane Doe |
| Bob Smith |
| Sally Jones |
| Mike Brown |
| Linda Davis |
| Tom Wilson |
| Sarah Taylor |
| Jack Clark |
| Amy White |
| Bill Walker |
| Karen Hall |
| Dave Carter |
| Julie Lee |
| Frank Adams |
| Ann Baker |

Result 9

4. SELECT * FROM donor_T WHERE last_donation < '2021-12-25';

```
832
833   /*This query retrieves the donor's information for donors that last donated before December 25, 2021.*/
834 •  SELECT * FROM donor_T WHERE last_donation < '2021-12-25';
835
836   /* This query returns the results of all the nurses who are included in the staff table of this database.*/
837 •  SELECT * FROM staff_T WHERE job_title = 'Nurse';
838
839   #fetch donor details
840 •  select * from donor_T;
841
842   #fetch patient details
```

Result Grid Filter Rows: Export: Wrap Cell Content:

| donor_id | first_name | last_name | email | phone | blood_type | gender | birth_date | weight | last_donation |
|---|---|---|---|---|---|---|---|---|---|
| 103 | Sally | Jones | sally.jones@example.com | 4445556666 | B+ | Female | 1998-04-04 | 140 | 2021-10-15 |
| 105 | Linda | Davis | linda.davis@example.com | 4445556666 | A+ | Female | 1979-06-06 | 130 | 2021-12-05 |
| 111 | Karen | Hall | karen.hall@example.com | 4445556666 | AB- | Female | 1990-12-12 | 140 | 2021-11-20 |
| 120 | Emily | Wilson | emily.wilson@example.com | 5556668888 | B- | Female | 1999-04-15 | 120 | 2021-11-10 |
| 126 | Jessica | Clark | jessica.clark@example.com | 5554445555 | O+ | Female | 1997-01-07 | 125 | 2021-10-20 |

5. SELECT * FROM staff_T WHERE job_title = 'Nurse';

```
579 •  SELECT * FROM staff_T WHERE job_title = 'Nurse';
```

100% 49:579

Result Grid Filter Rows: Search Export:

| staff_id | staff_first_na... | staff_last_name | staff_pho... | job_title | donation_center_id |
|---|---|---|---|---|---|
| 115 | Susan | Reel | 888-2544 | Nurse | 5 |
| 116 | Kate | Moss | 888-3521 | Nurse | 6 |
| 120 | Omar | Willow | 888-3248 | Nurse | 10 |
| 125 | Selena | Gomez | 888-8552 | Nurse | 15 |
| 130 | Emma | Lee | 888-2544 | Nurse | 20 |
| 131 | Noah | Garcia | 888-3521 | Nurse | 21 |
| 135 | Ava | Jackson | 888-3248 | Nurse | 25 |
| 140 | Michael | Martin | 888-8552 | Nurse | 30 |
| 145 | Aiden | Hall | 888-2544 | Nurse | 35 |
| 146 | Mason | Young | 888-3521 | Nurse | 36 |
| 150 | Elizabeth | Scott | 888-3248 | Nurse | 40 |
| 155 | Jayden | Carter | 888-8552 | Nurse | 45 |
| 160 | Alexander | Campbell | 888-2544 | Nurse | 50 |

staff_T 15

6. select * from donor_T;

```
1085
1086    #fetch donor details
1087 •  select * from donor_T;
1088
1089    #fetch patient details
1090 •  select * from recipient_T;
1091
```

100% 1:1086

Result Grid Filter Rows: Search Edit: Export/Import:

| donor_id | first_name | last_name | email | phone | blood_type | gender | birth_date | weight | last_donation |
|---|---|---|---|---|---|---|---|---|---|
| 102 | Bob | Smith | bob.smith@example.com | 1112223333 | AB- | Male | 1985-03-03 | 170 | NULL |
| 103 | Sally | Jones | sally.jones@example.com | 4445556666 | B+ | Female | 1998-04-04 | 140 | 2021-10-15 |
| 104 | Mike | Brown | mike.brown@example.com | 7778889999 | O- | Male | 1992-05-05 | 160 | 2022-02-20 |
| 105 | Linda | Davis | linda.davis@example.com | 4445556666 | A+ | Female | 1979-06-06 | 130 | 2021-12-05 |
| 106 | Tom | Wilson | tom.wilson@example.com | 5556667777 | B- | Male | 1996-07-07 | 155 | NULL |
| 107 | Sarah | Taylor | sarah.taylor@example.com | 8889990000 | AB+ | Female | 1988-08-08 | 145 | 2022-01-10 |
| 108 | Jack | Clark | jack.clark@example.com | 7778889999 | O- | Male | 1991-09-09 | 170 | 2022-03-25 |
| 109 | Amy | White | amy.white@example.com | 2223334444 | A- | Female | 1993-10-10 | 125 | NULL |
| 110 | Bill | Walker | bill.walker@example.com | 5556667777 | B+ | Male | 1986-11-11 | 165 | 2022-02-15 |
| 111 | Karen | Hall | karen.hall@example.com | 4445556666 | AB- | Female | 1990-12-12 | 140 | 2021-11-20 |
| 112 | Dave | Carter | dave.carter@example.com | 1112223333 | O+ | Male | 1997-01-13 | 175 | NULL |
| 113 | Julie | Lee | julie.lee@example.com | 7778889999 | A+ | Female | 1982-02-14 | 120 | 2022-03-05 |

7. select * from recipient_T;

```
#fetch patient details
select * from recipient_T;
```

100% 27:1090

Result Grid Filter Rows: Search Edit: Export/Import:

| recipient_id | first_name | last_name | email | phone | blood_type... | gender | age | address | city | state | zip_code |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1002 | Bob | Smith | bobsmith@example.com | 555-9012 | 3 | Male | 42 | 789 Oak St | Somewhere | FL | 34567 |
| 1003 | Alice | Johnson | alicejohnson@example.com | 555-3456 | 4 | Female | 52 | 321 Pine St | Nowhere | TX | 89012 |
| 1004 | Tom | Williams | tomwilliams@example.com | 555-7890 | 5 | Male | 20 | 654 Cedar St | Anytown | CA | 12345 |
| 1005 | Samantha | Brown | samanthabrown@example.com | 555-2345 | 6 | Female | 29 | 987 Maple St | Othertown | NY | 67890 |
| 1006 | David | Davis | daviddavis@example.com | 555-6789 | 7 | Male | 39 | 654 Cedar St | Somewhere | FL | 34567 |
| 1007 | Sarah | Jones | sarahjones@example.com | 555-0123 | 8 | Female | 48 | 321 Pine St | Nowhere | TX | 89012 |
| 1008 | Michael | Garcia | michaelgarcia@example.com | 555-4567 | 1 | Male | 25 | 123 Main St | Anytown | CA | 12345 |
| 1009 | Michelle | Martinez | michellemartinez@example.com | 555-8901 | 2 | Female | 31 | 456 Elm St | Othertown | NY | 67890 |
| 1010 | Jose | Hernandez | josehernandez@example.com | 555-2345 | 3 | Male | 43 | 789 Oak St | Somewhere | FL | 34567 |
| 1011 | Maria | Lopez | marialopez@example.com | 555-6789 | 4 | Female | 55 | 321 Pine St | Nowhere | TX | 89012 |
| 1012 | Christopher | Clark | christopherclark@example.com | 555-0123 | 5 | Male | 19 | 654 Cedar St | Anytown | CA | 12345 |
| 1013 | Ashley | Rodriguez | ashleyrodriguez@example.com | 555-4567 | 6 | Female | 27 | 987 Maple St | Othertown | NY | 67890 |

recipient_T 7

8. SELECT concat(first_name,' ',last_name) as 'Name', blood_type, gender from donor_T where last_donation IS NOT NULL group by donor_id;

```
#donors who have donated blood in the past
SELECT concat(first_name,' ',last_name) as 'Name', blood_type, gender from donor_T
where last_donation IS NOT NULL group by donor_id;
```

100% 51:1094

Result Grid Filter Rows: Search Export:

| Name | blood_type | gender |
|---|---|---|
| Jane Doe | O+ | Female |
| Sally Jones | B+ | Female |
| Mike Brown | O- | Male |
| Linda Davis | A+ | Female |
| Sarah Taylor | AB+ | Female |
| Jack Clark | O- | Male |
| Bill Walker | B+ | Male |
| Karen Hall | AB- | Female |
| Julie Lee | A+ | Female |
| Frank Adams | B- | Male |
| Ann Baker | AB+ | Female |
| Rob Miller | O+ | Male |
| John Smith | A+ | Male |

Result 8

9. SELECT COUNT(*) AS donor_count, blood_type FROM donor_T group by blood_type;

```
#number of donors available with respect to their blood_types to check blood supplies
#This information can be useful in a blood donation drive to help organizers plan
#donation campaigns and ensure that an adequate supply of blood is available for
#patients who need it
SELECT COUNT(*) AS donor_count, blood_type FROM donor_T
group by blood_type;
```

100% 21:1101

Result Grid Filter Rows: Search Export:

| donor_count | blood_type |
|---|---|
| 8 | A+ |
| 7 | O+ |
| 4 | AB- |
| 7 | B+ |
| 7 | O- |
| 7 | B- |
| 7 | AB+ |
| 5 | A- |

Result 9

10. select count(*) from donation_center_t where state = "TX";

```
-- Find the number of donation centers that are in Texas
select count(*) from donation_center_t where state = "TX";

-- Select all the donor's information that have O- as the blood type
```

100% 59:1104

Result Grid Filter Rows: Search Export:

| ... |
|---|
| 8 |

11. select * from labs_t where blood_type = "O-";

```
-- Select all the donor's information that have O- as the blood type
select * from labs_t where blood_type = "O-";
```

100% 46:1107

Result Grid Filter Rows: Search Edit: Export/Import:

| lab_id | donor_id | blood_ty... |
|---|---|---|
| 8 | 107 | O- |
| 16 | 115 | O- |
| 24 | 123 | O- |
| 32 | 131 | O- |
| 40 | 139 | O- |
| 48 | 147 | O- |
| NULL | NULL | NULL |

labs_t 11

12. Select state, count(*) AS count from donation_center_t group by state order by count desc;

```
-- Select the number of donation centers from each state and find out which state has more number of donation centers
Select state, count(*) AS count from donation_center_t group by state order by count desc;
```

100% 91:1110

Result Grid Filter Rows: Search Export:

| st | count |
|---|---|
| T | 8 |
| F | 5 |
| C | 4 |
| N | 3 |
| C | 3 |
| K | 2 |
| G | 2 |
| D | 2 |
| W | 2 |
| IL | 2 |
| M | 1 |
| M | 1 |
| T | 1 |

13. select last_name, count(*) AS TOTAL from donor_t group by last_name order by TOTAL desc;

```
-- Find the number of people from a each family that have donated blood
-- and the family that had more number of people that donated
select last_name, count(*) AS TOTAL from donor_t group by last_name order by TOTAL desc;

-- Find the amount of blood samples requested by a requester from lowest to highest
```

100% 89:1114

Result Grid Filter Rows: Search Export:

| last_name | TOTAL |
|---|---|
| Clark | 3 |
| Taylor | 3 |
| Brown | 3 |
| Garcia | 2 |
| Davis | 2 |
| Wilson | 2 |
| Smith | 2 |
| Doe | 2 |
| Walker | 2 |
| Hall | 2 |
| Baker | 2 |
| Anderson | 2 |
| Thomas | 2 |

Result 13

14. Select request_id, request_amount from REQUEST_T order by Request_Amount;

```
-- Find the amount of blood samples requested by a requester from lowest to highest
Select request_id, request_amount from REQUEST_T order by Request_Amount;

-- All the Requests post 4th April
```

100% 74:1117

Result Grid Filter Rows: Search Export:

| request_id | request_amou... |
|---|---|
| 44 | 50 |
| 36 | 50 |
| 28 | 50 |
| 45 | 60 |
| 9 | 60 |
| 48 | 90 |
| 46 | 90 |
| 19 | 90 |
| 16 | 90 |
| 26 | 100 |
| 49 | 100 |
| 42 | 100 |
| 34 | 100 |

REQUEST_T 14

15. select * FROM Request_T where Request_date > 04-04-2023;

```
-- All the Requests post 4th April
select * FROM Request_T where Request_date > 04-04-2023;

-- Arranging all the donation quantities in ascending order of their donation in the donation Table(Donation_T)
```

100% 57:1120

Result Grid Filter Rows: Search Edit: Export/Import:

| Request_ID | Requester_phone | Request_Amou... | Request_Date | Recipient_ID |
|---|---|---|---|---|
| 1 | 1234567890 | 150 | 2023-04-01 | 1000 |
| 2 | 1234567899 | 250 | 2023-02-01 | 1001 |
| 3 | 1234556676 | 150 | 2023-01-01 | 1002 |
| 4 | 1234556677 | 120 | 2022-01-02 | 1003 |
| 5 | 12345566767 | 180 | 2023-01-09 | 1004 |
| 6 | 12345566767 | 150 | 2023-09-09 | 1005 |
| 7 | 1122334455 | 150 | 2023-03-02 | 1006 |
| 8 | 6677889900 | 180 | 2023-08-01 | 1007 |
| 9 | 4455667788 | 60 | 2023-02-19 | 1008 |
| 10 | 3344556677 | 120 | 2023-03-19 | 1009 |
| 11 | 1122334455 | 150 | 2023-02-21 | 1010 |
| 12 | 2244668800 | 210 | 2023-01-20 | 1011 |
| 13 | 1122334567 | 180 | 2023-01-15 | 1012 |

Request_T 15

16. select blood_inventory_id, blood_type_id, blood_expiry_date, blood_storage_location, blood_category from blood_inventory_T where blood_expiry_date < '2022-04-01';

```
/* This query check for the blood details of donors whose blood expires before April 1st 2022 */
select blood_inventory_id, blood_type_id, blood_expiry_date, blood_storage_location, blood_category
from blood_inventory_T where blood_expiry_date < '2022-04-01';
```

100% 63:1156

Result Grid Filter Rows: Search Edit: Export/Import:

| blood_inventory... | blood_type... | blood_expiry_da... | blood_storage_locat... | blood_catego... |
|---|---|---|---|---|
| 1 | 1 | 2022-02-12 | FRIDGE 1 | A |
| 2 | 2 | 2022-03-16 | FRIDGE 2 | A |
| 8 | 8 | 2022-03-16 | FRIDGE 3 | A |
| 23 | 4 | 2022-03-29 | FRIDGE 2 | O |
| 24 | 5 | 2022-03-28 | FRIDGE 2 | O |
| 25 | 6 | 2022-03-27 | FRIDGE 2 | O |
| 26 | 7 | 2022-03-26 | FRIDGE 2 | O |
| 27 | 8 | 2022-03-25 | FRIDGE 2 | O |
| 28 | 1 | 2022-03-24 | FRIDGE 2 | O |
| 29 | 2 | 2022-03-23 | FRIDGE 2 | O |
| 30 | 3 | 2022-03-22 | FRIDGE 2 | O |
| 31 | 4 | 2022-03-21 | FRIDGE 2 | O |
| 32 | 5 | 2022-03-21 | FRIDGE 2 | O |

blood_inventory_T 1

17. select blood_category,blood_storage_location, min(blood_expiry_date) as 'Use 1st', max(blood_expiry_date) as 'Use Last'  from blood_inventory_T group by blood_category, blood_storage_location;

```
select blood_category,blood_storage_location, min(blood_expiry_date) as 'Use 1st', max(blood_expiry_date) as 'Use Last'
from blood_inventory_T group by blood_category, blood_storage_location;
```

100% 72:1162

Result Grid Filter Rows: Search Export:

| blood_catego... | blood_storage_locat... | Use 1st | Use Last |
|---|---|---|---|
| A | FRIDGE 1 | 2022-02-12 | 2022-07-01 |
| A | FRIDGE 2 | 2022-03-16 | 2022-06-12 |
| B | FRIDGE 1 | 2022-04-02 | 2022-04-08 |
| AB | FRIDGE 3 | 2022-02-04 | 2022-06-28 |
| 0 | FRIDGE 3 | 2022-06-23 | 2022-06-23 |
| A | FRIDGE 3 | 2022-03-16 | 2022-03-16 |
| B | FRIDGE 3 | 2022-04-21 | 2022-05-07 |
| O | FRIDGE 2 | 2022-03-21 | 2022-03-29 |
| AB | FRIDGE 1 | 2022-02-05 | 2022-07-22 |
| AB | FRIDGE 2 | 2022-02-03 | 2022-02-03 |
| A+ | FRIDGE 3 | 2022-07-03 | 2022-07-09 |
| A- | FRIDGE 2 | 2022-07-08 | 2022-07-08 |
| A+ | FRIDGE 1 | 2022-07-07 | 2022-07-07 |

18. select donor_id, eligibility_status from donor_medical_history_T where eligibility_status = 'Y';

```
/* This query gives us the donor IDs of donors who are eligible to donate blood */
select donor_id, eligibility_status from donor_medical_history_T where eligibility_status = 'Y';
```

100% 97:1165

Result Grid Filter Rows: Search Export:

| donor_id | eligibility_stat... |
|---|---|
| 106 | Y |
| 107 | Y |
| 111 | Y |
| 113 | Y |
| 114 | Y |
| 115 | Y |
| 117 | Y |
| 118 | Y |
| 119 | Y |
| 120 | Y |
| 123 | Y |
| 125 | Y |
| 126 | Y |

donor_medical_history_T 3

19. select donor_id, diseases, infections, covid_vaccination_status, eligibility_status from donor_medical_history_T where (infections = 'cough' and covid_vaccination_status =

'N');

```
/* This query shows if a person who has not taken a covid vaccination and is suffering from cough can donate blood or not.
The output shows that the person can donate blood */
select donor_id, diseases, infections, covid_vaccination_status, eligibility_status
from donor_medical_history_T where (infections = 'cough' and covid_vaccination_status = 'N');

/* This query tells us that a person suffering from Malaria, Ebola, HIV or Hepatits B cannot donate his/her blood*/
```

100% 94:1170

Result Grid Filter Rows: Search Export:

| donor_id | diseases | infections | covid_vaccination_sta... | eligibility_stat... |
|---|---|---|---|---|
| 123 | NULL | cough | N | Y |

donor_medical_history_T 4

Action Output

20. select donor_id, diseases, infections, covid_vaccination_status, eligibility_status from donor_medical_history_T where diseases in ('malaria','ebola','hiv','hepatitis b');

```
/* This query tells us that a person suffering from Malaria, Ebola, HIV or Hepatits B cannot donate his/her blood*/
select donor_id, diseases, infections, covid_vaccination_status, eligibility_status
from donor_medical_history_T where diseases in ('malaria','ebola','hiv','hepatitis b');
```

100% 88:1174

Result Grid Filter Rows: Search Export:

| donor_id | diseases | infections | covid_vaccination_sta... | eligibility_stat... |
|---|---|---|---|---|
| 100 | Malaria | NULL | Y | N |
| 103 | Ebola | NULL | Y | N |
| 105 | HIV | NULL | Y | N |
| 109 | Malaria | cough | Y | N |
| 110 | HIV | cough | Y | N |
| 112 | Ebola | NULL | Y | N |
| 116 | Hepatitis B | NULL | Y | N |
| 124 | Malaria | NULL | Y | N |
| 129 | HIV | NULL | Y | N |
| 133 | HIV | NULL | Y | N |
| 146 | HIV | NULL | Y | N |
| 140 | HIV | NULL | Y | N |

donor_medical_history_T 5

## Complex Queries:

1. SELECT CONCAT(d.first_name,' ', d.last_name) AS 'Donor Name',
   d.blood_type,CONCAT(d1.first_name, ' ', d1.last_name)
   AS 'Matching Donor Name', d1.blood_type
   FROM donor_T d
   JOIN donor_T d1 ON d.blood_type = d1.blood_type
   WHERE d.donor_id < d1.donor_id;

```
SELECT CONCAT(d.first_name,' ', d.last_name) AS 'Donor Name', d.blood_type,CONCAT(d1.first_name, ' ', d1.last_name)
AS 'Matching Donor Name', d1.blood_type
FROM donor_T d
JOIN donor_T d1 ON d.blood_type = d1.blood_type
WHERE d.donor_id < d1.donor_id;
```

100% 32:598

Result Grid Filter Rows: Search Export:

| Donor Name | blood_type | Matching Donor Name | blood_type |
|---|---|---|---|
| Johnny Doe | A+ | Linda Davis | A+ |
| Johnny Doe | A+ | Julie Lee | A+ |
| Johnny Doe | A+ | John Smith | A+ |
| Johnny Doe | A+ | Jennifer Garcia | A+ |
| Johnny Doe | A+ | Karen Turner | A+ |
| Johnny Doe | A+ | Evan King | A+ |
| Johnny Doe | A+ | Jennifer Garcia | A+ |
| Jane Doe | O+ | Dave Carter | O+ |
| Jane Doe | O+ | James Davis | O+ |
| Jane Doe | O+ | Jessica Clark | O+ |
| Jane Doe | O+ | Elizabeth Scott | O+ |
| Jane Doe | O+ | Jessica Clark | O+ |
| Bob Smith | AB- | Karen Hall | AB- |
| Bob Smith | AB- | Richard Roberts | AB- |
| Bob Smith | AB- | Richard Roberts | AB- |
| Sally Jones | B+ | Bill Walker | B+ |

Result 44

/* This query searches through the donor table to retrieve the findings of all pairs of donors who have the same blood type. The WHERE clause helps to ensure there are no duplicates.

2. SELECT center_name, address, city, state FROM donation_center_T
   WHERE donation_center_id IN
   (SELECT donation_center_id
   FROM staff_T
   WHERE job_title = "Doctor");

```
SELECT center_name, address, city, state FROM donation_center_T
WHERE donation_center_id IN
(SELECT donation_center_id
FROM staff_T
WHERE job_title = "Doctor");
```

100% 29:584

Result Grid Filter Rows: Search Export:

| center_name | address | city | state |
|---|---|---|---|
| LifeSouth Community Blood Centers | 5580 Heffernan Circle | Minneapolis | MN |
| UCLA Blood & Platelet Center | 75 Karstens Avenue | Mountain View | CA |
| Blood Bank of Hawaii | 9718 Lukken Terrace | Honolulu | HI |
| Bloodworks Northwest | 315 Anniversary Center | Beaverton | OR |
| Blood Bank of Arizona | 6 3rd Terrace | Phoenix | AZ |
| Gulf Coast Regional Blood Center | 09835 Kropf Crossing | Corpus Christi | TX |
| Memorial Blood Centers | 22 Pearson Crossing | Milwaukee | WI |
| Cascade Regional Blood Services | 00265 Surrey Junction | Seattle | WA |
| Northern California Community Blood Bank | 95308 2nd Plaza | San Jose | CA |
| Versiti Illinois | 952 Elmside Road | Chicago | IL |
| Bloodworks Northwest | 0 Crest Line Circle | Washington | DC |
| OneBlood | 6923 Basil Park | West Palm B... | FL |
| Arkansas Blood Institute | 74 Milwaukee Junction | Anchorage | AK |

/*This subquery retrieves the donation center name, address, city, and state of the donation center where there is at least one doctor present as staff.

3. -- Get the name of the donor and the blood type
use blooddonationdb;
SELECT donor_t.last_name, blood_types_t.blood_type_name
FROM donor_t CROSS JOIN blood_types_t;

```
-- Get the first name of the donor and the blood type
use blooddonationdb;
SELECT
    donor_t.last_name, blood_types_t.blood_type_name
FROM
    donor_t
        CROSS JOIN
    blood_types_t;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content

| last_name | blood_type_name |
|---|---|
| Doe | B+ |
| Doe | A- |
| Doe | A+ |
| Doe | O- |
| Doe | O+ |
| Doe | AB- |
| Doe | AB+ |
| Doe | B- |
| Doe | B+ |

4. -- Get the last name of the donor and the blood type
SELECT donor_t.last_name, blood_types_t.blood_type_name FROM donor_t CROSS JOIN blood_types_t;

```
-- Get the last name of the donor and the blood type
SELECT
    donor_t.last_name, blood_types_t.blood_type_name
FROM
    donor_t
CROSS JOIN
    blood_types_t;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Conte

| last_name | blood_type_name |
|---|---|
| Doe | B+ |
| Doe | A- |
| Doe | A+ |
| Doe | O- |
| Doe | O+ |
| Doe | AB- |

5. SELECT donor_T.donor_id, concat(donor_T.first_name,' ', donor_T.last_name) as 'Full Name', donor_T.phone, donor_T.blood_type, donor_T.gender, donor_T.birth_date, donor_T.weight, donor_T.last_donation, donor_medical_history.diseases, donor_medical_history.infections, donor_medical_history.covid_vaccination_status AS 'Covid Vaccinated?', donor_medical_history.eligibility_status FROM donor_T INNER JOIN donor_medical_history ON donor_T.donor_id = donor_medical_history.donor_id;

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| donor_id | Full Name | phone | blood_type | gender | birth_date | weight | last_donation | diseases | infections | Covid Vaccinated? | eligibility_status |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 100 | Johnny Doe | 1234567890 | A+ | Male | 1990-01-01 | 150 | NULL | Malaria | NULL | Y | N |
| 103 | Sally Jones | 4445556666 | B+ | Female | 1998-04-04 | 140 | 2021-10-15 | Ebola | NULL | Y | N |
| 104 | Mike Brown | 7778889999 | O- | Male | 1992-05-05 | 160 | 2022-02-20 | NULL | NULL | Y | Y |
| 105 | Linda Davis | 4445556666 | A+ | Female | 1979-06-06 | 130 | 2021-12-05 | HIV | NULL | Y | N |
| 107 | Sarah Taylor | 8889990000 | AB+ | Female | 1988-08-08 | 145 | 2022-01-10 | NULL | NULL | N | Y |
| 112 | Dave Carter | 1112223333 | O+ | Male | 1997-01-13 | 175 | NULL | Ebola | NULL | Y | N |
| 113 | Julie Lee | 7778889999 | A+ | Female | 1982-02-14 | 120 | 2022-03-05 | NULL | NULL | Y | Y |
| 116 | Rob Miller | 5556667777 | O | Male | 1994-03-15 | 160 | 2022-01-01 | Hepatiti... | NULL | Y | N |
| 124 | Melanie And... | 5551112222 | B+ | Female | 1994-08-20 | 150 | 2022-03-15 | Malaria | NULL | Y | N |
| 126 | Jessica Clark | 5554445555 | O+ | Female | 1997-01-07 | 125 | 2021-10-20 | NULL | NULL | Y | Y |
| 127 | Richard Rob... | 5556667777 | AB- | Male | 1981-07-10 | 175 | 2022-01-05 | NULL | NULL | Y | Y |
| 131 | Matthew Co... | 5554445555 | AB+ | Male | 1978-09-05 | 200 | 2022-02-25 | NULL | NULL | Y | Y |
| 137 | Lily Allen | 444-555-6666 | O- | Female | 1988-12-30 | 150 | 2021-10-15 | NULL | NULL | Y | Y |

/*This will give you a result set that combines the columns from both tables where there is a matching donor_id. The INNER JOIN keyword will only return rows where there is a match in both tables.*/

6. SELECT * FROM blood_inventory NATURAL JOIN blood_types_T;

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| blood_type_id | blood_inventory_id | blood_expiry_date | blood_storage_location | blood_category | blood_type_name |
|---|---|---|---|---|---|
| 1 | 1 | 2022-02-12 | FRIDGE 1 | A | A+ |
| 2 | 2 | 2022-03-16 | FRIDGE 2 | A | A- |
| 3 | 3 | 2022-05-24 | FRIDGE 2 | A | B+ |
| 4 | 4 | 2022-06-12 | FRIDGE 2 | A | B- |
| 5 | 5 | 2022-04-02 | FRIDGE 1 | B | AB+ |
| 6 | 6 | 2022-06-28 | FRIDGE 3 | AB | AB- |
| 7 | 7 | 2022-06-23 | FRIDGE 3 | 0 | O+ |
| 8 | 8 | 2022-03-16 | FRIDGE 3 | A | O- |
| 9 | 9 | 2022-05-14 | FRIDGE 1 | A | A+ |
| 10 | 10 | 2022-07-01 | FRIDGE 1 | A | A- |
| 11 | 11 | 2022-04-19 | FRIDGE 2 | A | B+ |
| 12 | 12 | 2022-04-08 | FRIDGE 1 | B | B- |
| 13 | 13 | 2022-05-07 | FRIDGE 3 | B | AB+ |
| 14 | 14 | 2022-04-29 | FRIDGE 3 | B | AB- |

/*This query selects all columns from both tables and uses the NATURAL JOIN clause to join the two tables on the blood_type_id column. The resulting table will have columns for blood_inventory_id, blood_expiry_date, blood_storage_location, blood_category, blood_type_id, and blood_type_name.*/

7. SELECT * FROM blood_types_T LEFT OUTER JOIN blood_inventory ON blood_types_T.blood_type_id = blood_inventory.blood_type_id;

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| blood_type_id | blood_type_name | blood_inventory_id | blood_type_id | blood_expiry_date | blood_storage_location | blood_category |
|---|---|---|---|---|---|---|
| 1 | A+ | 1 | 1 | 2022-02-12 | FRIDGE 1 | A |
| 2 | A- | 2 | 2 | 2022-03-16 | FRIDGE 2 | A |
| 3 | B+ | 3 | 3 | 2022-05-24 | FRIDGE 2 | A |
| 4 | B- | 4 | 4 | 2022-06-12 | FRIDGE 2 | A |
| 5 | AB+ | 5 | 5 | 2022-04-02 | FRIDGE 1 | B |
| 6 | AB- | 6 | 6 | 2022-06-28 | FRIDGE 3 | AB |
| 7 | O+ | 7 | 7 | 2022-06-23 | FRIDGE 3 | 0 |
| 8 | O- | 8 | 8 | 2022-03-16 | FRIDGE 3 | A |
| 9 | A+ | 9 | 9 | 2022-05-14 | FRIDGE 1 | A |
| 10 | A- | 10 | 10 | 2022-07-01 | FRIDGE 1 | A |
| 11 | B+ | 11 | 11 | 2022-04-19 | FRIDGE 2 | A |
| 12 | B- | 12 | 12 | 2022-04-08 | FRIDGE 1 | B |
| 13 | AB+ | 13 | 13 | 2022-05-07 | FRIDGE 3 | B |
| 14 | AB- | 14 | 14 | 2022-04-29 | FRIDGE 3 | B |

Result 10

/*This query selects all columns from both tables and uses the LEFT OUTER JOIN clause to join the two tables on the blood_type_id column. The resulting table will have columns for blood_type_id, blood_type_name, blood_inventory_id, blood_expiry_date, blood_storage_location, and blood_category. If there is no matching row in the blood_inventory table, the columns for blood_inventory_id, blood_expiry_date, blood_storage_location, and blood_category will be filled with null values.*/

Functions:

1. -- function to fetch blood type using blood_type_id

```
DROP FUNCTION IF EXISTS GET_DONOR_BLOODTYPE;
DELIMITER //
CREATE FUNCTION GET_DONOR_BLOODTYPE
(
BLOOD_TYPE_ID_PARAM INT
)
RETURNS VARCHAR(5)
DETERMINISTIC
BEGIN
DECLARE BLOOD_TYPE_VAR VARCHAR(5);
SELECT blood_type_name INTO BLOOD_TYPE_VAR
FROM blood_types_T
WHERE BLOOD_TYPE_ID = BLOOD_TYPE_ID_PARAM;
RETURN BLOOD_TYPE_VAR;
END //
DELIMITER ;
SELECT GET_DONOR_BLOODTYPE('3');
```

```
CREATE FUNCTION GET_DONOR_BLOODTYPE
(
BLOOD_TYPE_ID_PARAM INT
)
RETURNS VARCHAR(5)
DETERMINISTIC
BEGIN
DECLARE BLOOD_TYPE_VAR VARCHAR(5);
SELECT blood_type_name INTO BLOOD_TYPE_VAR
FROM blood_types_T
WHERE BLOOD_TYPE_ID = BLOOD_TYPE_ID_PARAM;
RETURN BLOOD_TYPE_VAR;
END //
DELIMITER ;
SELECT GET_DONOR_BLOODTYPE('3');
```

100% 33:1058

Result Grid Filter Rows: Search Export:

| GET_DONOR_BLOODTYPE('3') |
| --- |
| B+ |

2. /*This function is used to find the staff ID of staff members using their first names and then selects all information for that staff member from the "staff_T" table using the retrieved staff ID.*/

```
DELIMITER //
CREATE FUNCTION get_staff_id
(
staff_first_name_param VARCHAR(50)
)
RETURNS int
DETERMINISTIC
BEGIN
DECLARE staff_id_var INT;
SELECT staff_id INTO staff_id_var FROM staff_T
WHERE staff_first_name = staff_first_name_param;
RETURN (staff_id_var);
END //
DELIMITER ;
SELECT get_staff_id('Abigail');
```

```
/*This function is used to find the staff ID of staff members using their first names and then selects all
 information for that staff member from the "staff_T" table using the retrieved staff ID.*/
DELIMITER //
CREATE FUNCTION get_staff_id
(
staff_first_name_param VARCHAR(50)
)
RETURNS int
DETERMINISTIC
BEGIN
DECLARE staff_id_var INT;
SELECT staff_id INTO staff_id_var FROM staff_T
WHERE staff_first_name = staff_first_name_param;
RETURN (staff_id_var);
END //
DELIMITER ;
SELECT get_staff_id('Abigail');
```

| get_staff_id('Abigail') |
|---|
| 537 |

3. /*This function is used to find the hospital ID using hospital names and then selects all information for that hospital from the "hospital_T" table using the retrieved hospital ID.*/

```
DELIMITER //
CREATE FUNCTION get_hospital_id
(
hospital_name_param VARCHAR(50)
)
RETURNS int
DETERMINISTIC
BEGIN
DECLARE hospital_id_var INT;
SELECT hospital_id INTO hospital_id_var FROM hospital_T
WHERE hospital_name = hospital_name_param;
RETURN (hospital_id_var);
END //
DELIMITER ;
SELECT get_hospital_id("Tampa General Hospital");
```

```
/*This function is used to find the hospital ID using hospital names and then selects all
 information for that hospital from the "hospital_T" table using the retrieved hospital ID.*/
DELIMITER //
CREATE FUNCTION get_hospital_id
(
hospital_name_param VARCHAR(50)
)
RETURNS int
DETERMINISTIC
BEGIN
DECLARE hospital_id_var INT;
SELECT hospital_id INTO hospital_id_var FROM hospital_T
WHERE hospital_name = hospital_name_param;
RETURN (hospital_id_var);
END //
DELIMITER ;
SELECT get_hospital_id("Tampa General Hospital");
```

| get_hospital_id("Tampa General Hospital") |
|---|
| 37 |

Views:

-- View all donor and recipient blood types matches

```
-- View all donor and recipient blood types matches
DROP VIEW if exists donor_recipient_bloodtype_V;
CREATE VIEW donor_recipient_bloodtype_V AS
SELECT d.donor_id, d.first_name AS donor_first_name, d.last_name AS donor_last_name, d.email AS donor_email,
       r.recipient_id, r.first_name AS recipient_first_name, r.last_name AS recipient_last_name, r.email AS reci
       b.blood_type_name
FROM donor_T AS d
JOIN recipient_T AS r ON d.blood_type = r.blood_type
JOIN blood_types_T AS b ON d.blood_type = b.blood_type_name;
select * from donor_recipient_bloodtype_V;

######stored procedures######

-- Donors who have donated blood in the past to figure out potential candidates
```

| donor_id | donor_first_name | donor_last_name | donor_email | recipient_id | recipient_first_name | recipient_last_name | recipient_email | blood_type_name |
|---|---|---|---|---|---|---|---|---|
| 100 | Johnny | Doe | john.doe@example.com | 1049 | Madison | Lee | m.lee@email.com | A+ |
| 100 | Johnny | Doe | john.doe@example.com | 1048 | Daniel | Lewis | d.lewis@email.com | A+ |
| 100 | Johnny | Doe | john.doe@example.com | 1005 | Samantha | Brown | samanthabrown@example.com | A+ |
| 100 | Johnny | Doe | john.doe@example.com | 1004 | Tom | Williams | tomwilliams@example.com | A+ |
| 100 | Johnny | Doe | john.doe@example.com | 1003 | Alice | Johnson | alicejohnson@example.com | A+ |

donor_recipient_bloodtype_V 53

Stored Procedures:

1. -- Donors who have donated blood in the past to figure out potential candidates

-- for future donations

```
DROP PROCEDURE IF EXISTS past_donor_info;

DELIMITER //
CREATE PROCEDURE past_donor_info()
BEGIN

SELECT concat(first_name,' ',last_name) as 'Name', blood_type, gender from donor_T
where last_donation IS NOT NULL group by donor_id;

END //
DELIMITER ;
CALL past_donor_info();
```

```
839 •  DROP PROCEDURE IF EXISTS past_donor_info;
840
841    DELIMITER //
842 •  CREATE PROCEDURE past_donor_info()
843    BEGIN
844
845    SELECT concat(first_name,' ',last_name) as 'Name', blood_type, gender from donor_T
846    where last_donation IS NOT NULL group by donor_id;
847
848    END //
849    DELIMITER ;
850 •  CALL past_donor_info();
851
852    -- number of donors available wrt their blood types for future inventory management
```

100% 24:850

Result Grid Filter Rows: Search Export:

| Name | blood_type | gender |
|---|---|---|
| Jane Doe | O+ | Female |
| Sally Jones | B+ | Female |
| Mike Brown | O- | Male |
| Linda Davis | A+ | Female |
| Sarah Taylor | AB+ | Female |
| Jack Clark | O- | Male |
| Bill Walker | B+ | Male |
| Karen Hall | AB- | Female |
| Julie Lee | A+ | Female |
| Frank Adams | B- | Male |
| Ann Baker | AB+ | Female |
| Rob Miller | O+ | Male |
| John Smith | A+ | Male |

Result 9

2. -- number of donors available wrt their blood_types for future inventory management
   -- to be prepared for unforeseen situations

DROP PROCEDURE IF EXISTS donor_count;

DELIMITER //
CREATE PROCEDURE donor_count()
BEGIN
SELECT COUNT(*) AS donor_count, blood_type
FROM donor_T
group by blood_type;
END //
DELIMITER ;
CALL donor_count();

```
855 •  DROP PROCEDURE IF EXISTS donor_count;
856
857    DELIMITER //
858 •  CREATE PROCEDURE donor_count()
859    BEGIN
860    SELECT COUNT(*) AS donor_count, blood_type
861    FROM donor_T
862    group by blood_type;
863    END //
864    DELIMITER ;
865 •  CALL donor_count();
866
```

100% 20:865

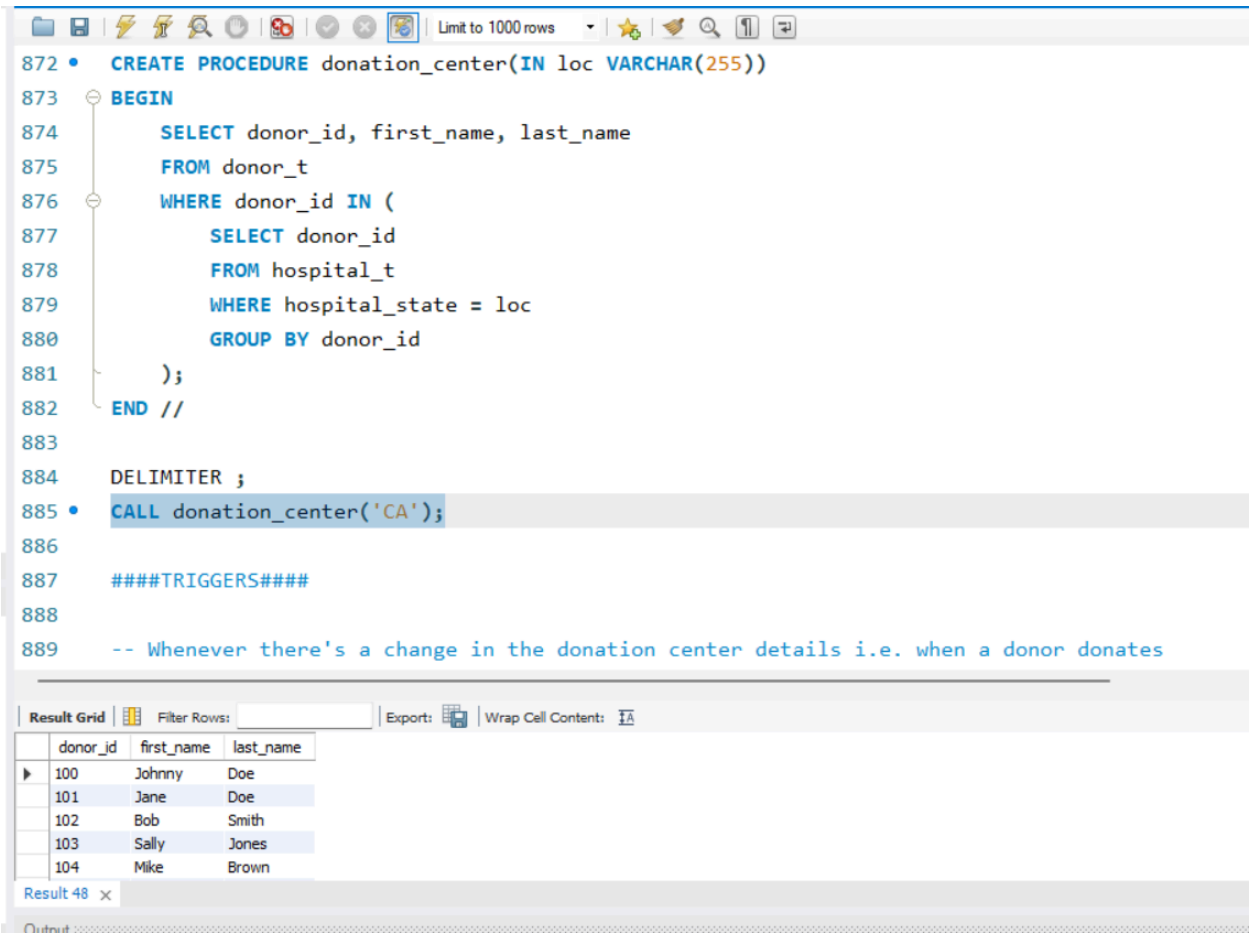
Result Grid Filter Rows: Search Export:

| donor_count | blood_type |
|---|---|
| 8 | A+ |
| 7 | O+ |
| 4 | AB- |
| 7 | B+ |
| 7 | O- |
| 7 | B- |
| 7 | AB+ |
| 5 | A- |

3. -- Select the donor id, first and last name of donors where their blood donated hospital
   -- is situated in a particular state

DROP PROCEDURE IF EXISTS donation_center;

```
DELIMITER //

CREATE PROCEDURE donation_center(IN loc VARCHAR(255))
BEGIN
   SELECT donor_id, first_name, last_name
   FROM donor_t
   WHERE donor_id IN (
      SELECT donor_id
      FROM hospital_t
      WHERE hospital_state = loc
      GROUP BY donor_id
   );
END //

DELIMITER ;
CALL donation_center('CA');
```




Triggers

1. -- Whenever there's a change in the donation center details i.e. when a donor donates
   -- blood again the donation center details and the staff details associated with him
   -- changes and at this event a BEFORE update trigger is invoked and the old location
   -- and staff details is stored in the audit tables for future references

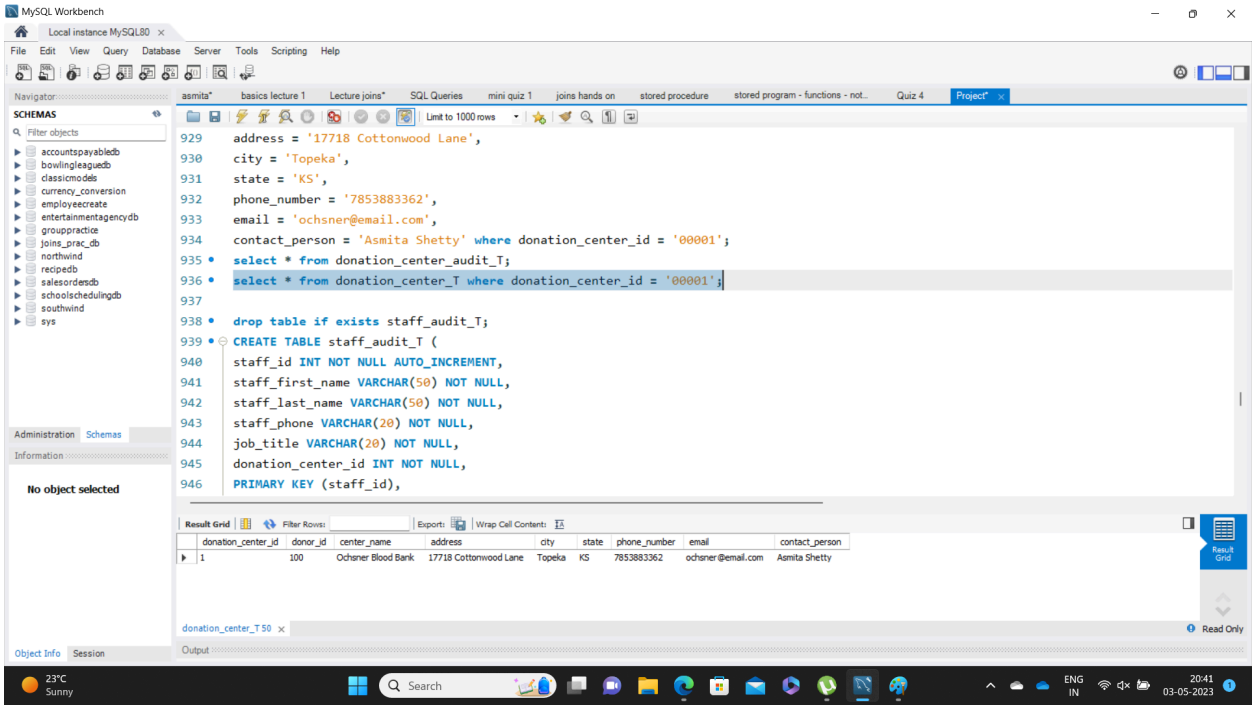
```
DROP table if exists donation_center_audit_T;
CREATE TABLE donation_center_audit_T (
  donation_center_id INT PRIMARY KEY,
  donor_id int not null,
  center_name VARCHAR(200) NOT NULL,
  address VARCHAR(100) NOT NULL,
  city VARCHAR(50) NOT NULL,
  state VARCHAR(2) NOT NULL,
  phone_number VARCHAR(10),
  email VARCHAR(200),
  contact_person VARCHAR(200),
  FOREIGN KEY (donor_id) REFERENCES donor_T (donor_id),
  changedat DATETIME DEFAULT NULL,
  action VARCHAR(50) DEFAULT NULL
);

DROP TRIGGER IF EXISTS before_donation_center_update;
CREATE TRIGGER before_donation_center_update
BEFORE UPDATE ON donation_center_T
FOR EACH ROW
INSERT INTO donation_center_audit_T
SET action = 'update',
donation_center_id = OLD.donation_center_id,
donor_id = OLD.donor_id,
center_name = OLD.center_name,
address = OLD.address,
city = OLD.city,
state = OLD.state,
phone_number= OLD.phone_number,
email = OLD.email,
contact_person = OLD.contact_person,
changedat = NOW();

update donation_center_T set
center_name = 'Ochsner Blood Bank',
address = '17718 Cottonwood Lane',
city = 'Topeka',
state = 'KS',
phone_number = '7853883362',
```

email = 'ochsner@email.com',
contact_person = 'Asmita Shetty' where donation_center_id = '00001';
select * from donation_center_audit_T;

```
923     email = OLD.email,
924     contact_person = OLD.contact_person,
925     changedat = NOW();
926
927     update donation_center_T set
928     center_name = 'Ochsner Blood Bank',
929     address = '17718 Cottonwood Lane',
930     city = 'Topeka',
931     state = 'KS',
932     phone_number = '7853883362',
933     email = 'ochsner@email.com',
934     contact_person = 'Asmita Shetty' where donation_center_id = '00001';
935     select * from donation_center_audit_T;
936     select * from donation_center_T where donation_center_id = '00001';
937
938     drop table if exists staff_audit_T;
939     CREATE TABLE staff_audit_T (
940     staff_id INT NOT NULL AUTO_INCREMENT,
```

| donation_center_id | donor_id | center_name | address | city | state | phone_number | email | contact_person | changedat | action |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 100 | LifeSouth Community Blood Centers | 5580 Heffernan Circle | Minneapolis | MN | 6124057748 | lifesouth@email.com | Nazifah Kamin | 2023-05-03 20:26:29 | update |

select * from donation_center_T where donation_center_id = '00001';




drop table if exists staff_audit_T;
CREATE TABLE staff_audit_T (

```
staff_id INT NOT NULL AUTO_INCREMENT,
staff_first_name VARCHAR(50) NOT NULL,
staff_last_name VARCHAR(50) NOT NULL,
staff_phone VARCHAR(20) NOT NULL,
job_title VARCHAR(20) NOT NULL,
donation_center_id INT NOT NULL,
PRIMARY KEY (staff_id),
FOREIGN KEY (donation_center_id) REFERENCES donation_center_T (donation_center_id),
changedat DATETIME DEFAULT NULL,
action VARCHAR(50) DEFAULT NULL
);

DROP TRIGGER IF EXISTS before_donation_center_update_staff;
CREATE TRIGGER before_donation_center_update_staff
BEFORE UPDATE ON staff_T
FOR EACH ROW
INSERT INTO staff_audit_T
SET action = 'update',
donation_center_id = OLD.donation_center_id,
staff_first_name = OLD.staff_first_name,
staff_last_name = OLD.staff_last_name,
job_title = OLD.job_title,
staff_phone = OLD.staff_phone,
changedat = NOW();

update staff_T set
staff_first_name = 'Asmita',
staff_last_name = 'Shetty',
staff_phone = '878-1234'
where donation_center_id ='00001';
select * from staff_audit_T;
```

```
962    staff_phone = OLD.staff_phone,
963    changedat = NOW();
964
965 •  update staff_T set
966    staff_first_name = 'Asmita',
967    staff_last_name = 'Shetty',
968    staff_phone = '878-1234'
969    where donation_center_id ='00001';
970 •  select * from staff_audit_T;
971    #updated
972 •  select * from staff_t where donation_center_id = '00001';
973
974
975    -- Now, we are going to create a delete trigger for the labs_t table.
976    -- When a lab is removed from our system, we can delete their record and then store it in our audit table.
977    -- Create audit table.
978 •  CREATE TABLE IF NOT EXISTS labs_T_Audit_1 (
979      lab_id INT,
```

| staff_id | staff_first_name | staff_last_name | staff_phone | job_title | donation_center_id | changedat | action |
|---|---|---|---|---|---|---|---|
| 1 | Nazifah | Kamin | 888-1234 | Doctor | 1 | 2023-05-03 20:26:29 | update |

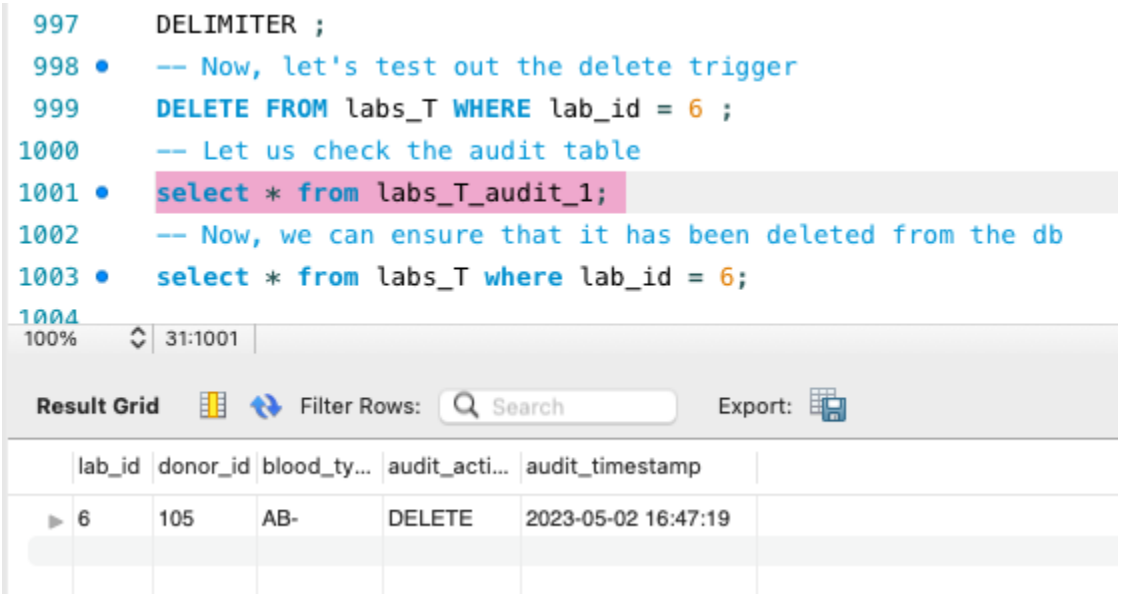
#updated
select * from staff_t where donation_center_id = '00001';

```
964 •  update staff_T set
965    staff_first_name = 'Asmita',
966    staff_last_name = 'Shetty',
967    staff_phone = '878-1234'
968    where donation_center_id ='00001';
969 •  select * from staff_audit_T;
970    #updated
971 •  select * from staff_t where donation_center_id = '00001';
```

| staff_id | staff_first_name | staff_last_name | staff_pho... | job_title | donation_center_id |
|---|---|---|---|---|---|
| 501 | Asmita | Shetty | 878-1234 | Doctor | 1 |
| NULL | NULL | NULL | NULL | NULL | NULL |

2. -- Now, we are going to create a delete trigger for the labs_t table.
   -- When a lab is removed from our system, we can delete their record and then store it in our audit table.

```
-- Create audit table.
CREATE TABLE IF NOT EXISTS labs_T_Audit_1 (
  lab_id INT,
  donor_id INT,
  blood_type VARCHAR(10),
  audit_action VARCHAR(10),
  audit_timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Now, we create the delete trigger
-- Create trigger for AFTER delete
DROP TRIGGER IF EXISTS labs_T_Delete_After;
DELIMITER //
CREATE TRIGGER labs_T_Delete_After
AFTER DELETE ON labs_T
FOR EACH ROW
BEGIN
  INSERT INTO labs_T_Audit_1 (lab_id, donor_id, blood_type, audit_action)
  VALUES (OLD.lab_id, OLD.donor_id, OLD.blood_type, 'DELETE');
END //

DELIMITER ;
-- Now, let's test out the delete trigger
DELETE FROM labs_T WHERE lab_id = 6 ;
-- Let us check the audit table
select * from labs_T_audit_1;
```




```
-- Now, we can ensure that it has been deleted from the db
select * from labs_T where lab_id = 6;
```

```
997     DELIMITER ;
998     -- Now, let's test out the delete trigger
999     DELETE FROM labs_T WHERE lab_id = 6 ;
1000    -- Let us check the audit table
1001    select * from labs_T_audit_1;
1002    -- Now, we can ensure that it has been deleted from the db
1003    select * from labs_T where lab_id = 6;
1004
```

100% 31:1001

Result Grid Filter Rows: Search Export:

| lab_id | donor_id | blood_ty... | audit_acti... | audit_timestamp |
|---|---|---|---|---|
| 6 | 105 | AB- | DELETE | 2023-05-02 16:47:19 |

3. -- Create audit table for labs_T after an insert of new lab

```
Drop table if exists labs_T_Audit_2;
CREATE TABLE IF NOT EXISTS labs_T_Audit_2 (
  lab_id INT,
  donor_id INT,
  blood_type VARCHAR(10),
  audit_action VARCHAR(10),
  audit_timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
-- Create the insert trigger
DROP TRIGGER IF EXISTS labs_T_Insert;
DELIMITER //
CREATE TRIGGER labs_T_Insert
AFTER INSERT ON labs_T
FOR EACH ROW
BEGIN
  INSERT INTO labs_T_Audit_2 (lab_id, donor_id, blood_type, audit_action)
  VALUES (NEW.lab_id, NEW.donor_id, NEW.blood_type, 'INSERT');
END //
DELIMITER ;
-- Insert a record into the labs_T table
INSERT INTO labs_T (lab_id, donor_id, blood_type) VALUES (90, 110, 'O+');

-- Query the labs_T_Audit table to check if the trigger has captured the new record
SELECT * FROM labs_T_Audit_2 WHERE lab_id = 90;
```

```
AFTER INSERT ON labs_T
FOR EACH ROW
BEGIN
  INSERT INTO labs_T_Audit_2 (lab_id, donor_id, blood_type, audit_action)
  VALUES (NEW.lab_id, NEW.donor_id, NEW.blood_type, 'INSERT');
END //
DELIMITER ;
-- Insert a record into the labs_T table
INSERT INTO labs_T (lab_id, donor_id, blood_type) VALUES (90, 110, 'O+');

-- Query the labs_T_Audit table to check if the trigger has captured the new record
SELECT * FROM labs_T_Audit_2 WHERE lab_id = 90;

-- Query the labs_T table to verify that the record has been inserted
SELECT * FROM labs_T WHERE lab_id = 90;
```

100% 40:1033

Result Grid Filter Rows: Search Edit: Export/Import:

| lab_id | donor_id | blood_ty... |
|---|---|---|
| 90 | 110 | O+ |
| NULL | NULL | NULL |

-- Now, we can ensure that it has been deleted from the db
SELECT * FROM labs_T WHERE lab_id = 90;

Limit to 1000 rows

```
  VALUES (OLD.lab_id, OLD.donor_id, OLD.blood_type, 'DELETE');
END //

DELIMITER ;
-- Now, let's test out the delete trigger
DELETE FROM labs_T WHERE lab_id = 6 ;
-- Let us check the audit table
select * from labs_T_audit_1;
-- Now, we can ensure that it has been deleted from the db
select * from labs_T where lab_id = 6;

-- Create audit table for labs_T after an insert of new lab
Drop table if exists labs_T_Audit_2;
CREATE TABLE IF NOT EXISTS labs_T_Audit_2 (
  lab_id INT,
  donor_id INT,
  blood_type VARCHAR(10),
  audit_action VARCHAR(10),
```

Result Grid Filter Rows: Export: Wrap Cell Content:

| lab_id | donor_id | blood_type |
|---|---|---|