

# BLOOD BANK +

*CIS 340-Database System*

*Blood Bank Database Management System*

*Project Members :*

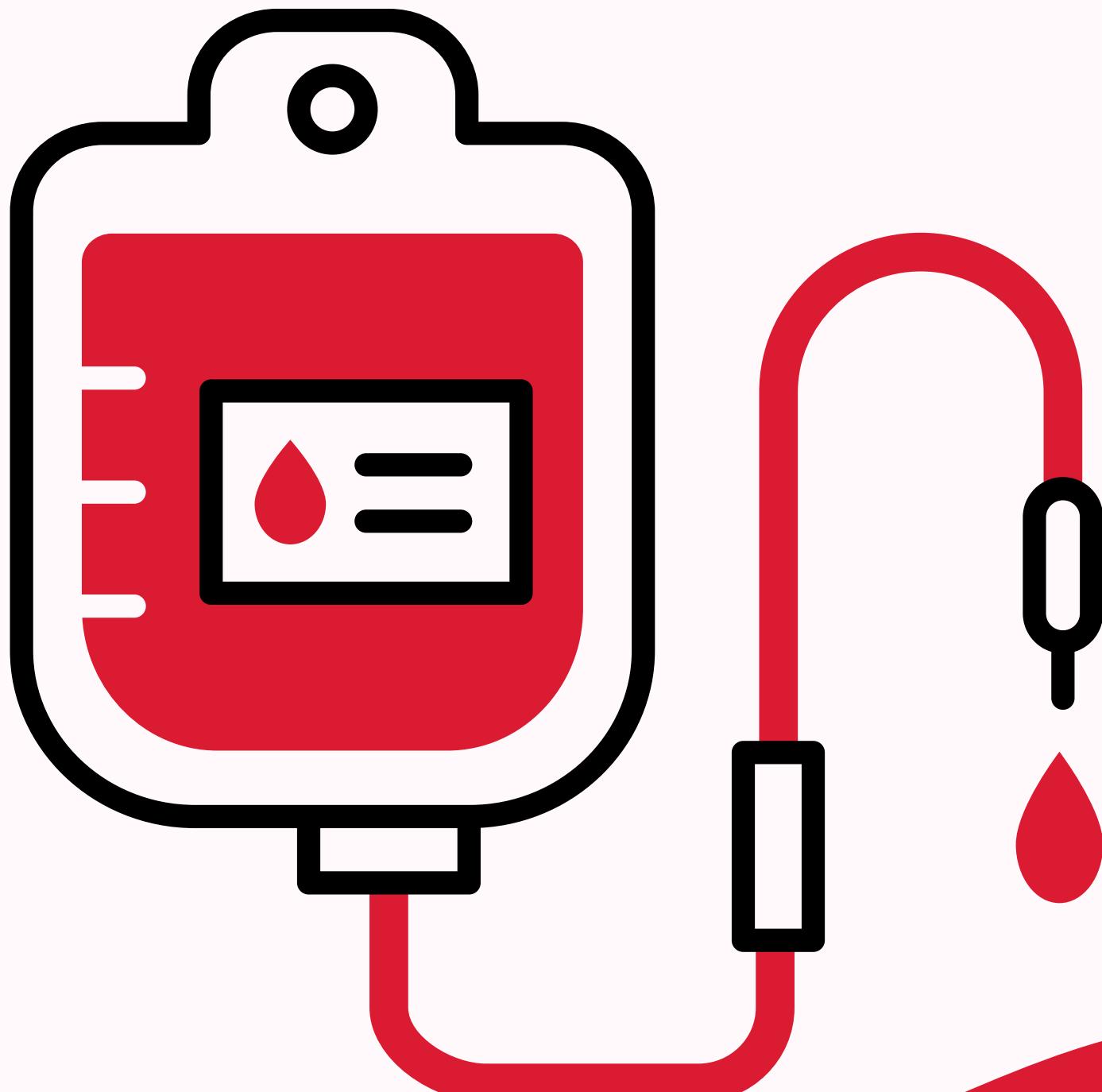
*Reem Eid Alatwi (431002766)*

*Almaha mufleh albalawi (431001421)*

*Shahad Ahmed Albalawi (431007856)*

*Fahdah Faheed Alrashidi (431000732)*

*Ghadeer Mohammed Alatwi (431008681)*



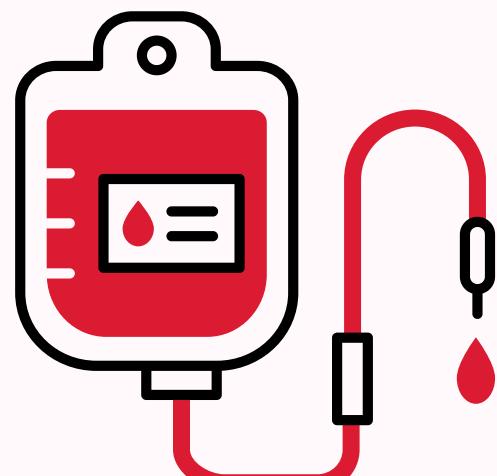
# INTRODUCTION

## *Revolutionizing Blood Management Through Automation*

In the ever-evolving landscape of healthcare, the efficiency and accuracy of blood management stand as critical pillars in ensuring patient well-being. Traditional manual processes within blood banks often lead to inefficiencies and delays, hindering the seamless flow of life-saving resources. Recognizing this challenge, our initiative seeks to transform blood management through the implementation of a sophisticated database system. By harnessing the power of automation, we aim to streamline and optimize every facet of the blood management process, from donor registration to distribution, with meticulous attention to detail and adherence to stringent quality standards.

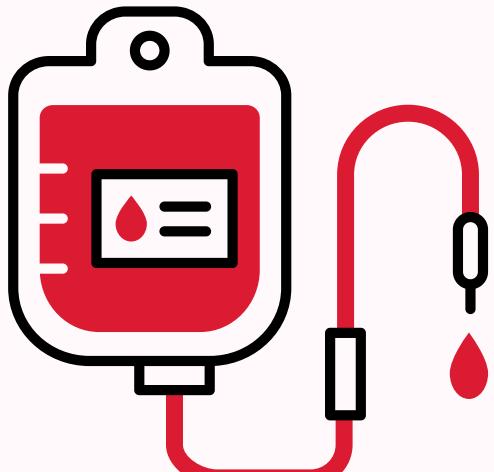
### Motivation: Enhancing Efficiency and Saving Lives

The motivation behind our endeavor stems from a deep-rooted commitment to improving healthcare outcomes and maximizing the impact of blood donation. Manual processes, characterized by paperwork and logistical hurdles, often result in delays and inefficiencies that can have grave consequences for patients in need. By automating key aspects of blood management, we not only aim to minimize these delays but also ensure the seamless flow of blood from donors to recipients. Ultimately, our goal is to enhance efficiency, reduce wastage, and most importantly, save lives by revolutionizing the way blood resources are managed and distributed within the healthcare system.



# KEY OBJECTIVES

- Efficient Donor Registration: Implementing a user-friendly registration system to facilitate seamless donor registration and data entry, ensuring accuracy and completeness of donor information.
- Comprehensive Blood Data Management: Establishing a robust database to securely store and manage blood data, including donor profiles, blood type information, and test results, enabling efficient retrieval and analysis when needed.
- Automated Blood Analysis: Relies on the use of the latest analytical devices to facilitate blood analysis procedures, enabling rapid and accurate evaluation of blood compatibility and quality. The critical analyst can securely save and retrieve their data for future use.
- Streamlined Patient Registration: Enabling healthcare providers to register patients in need of blood transfusions quickly and accurately, ensuring timely access to life saving treatments.
- Efficient Blood Distribution: Facilitating seamless communication between healthcare facilities and blood banks to expedite the distribution process, ensuring that blood reaches patients in a timely manner.
- Effective Management and Oversight: Providing managers with the tools and insights needed to effectively oversee blood bank operations, including inventory management, resource allocation, and quality control.



# MYSQL CODE OF BLOOD BANK DATABASE CREATION

MySQL code of blood bank database creation

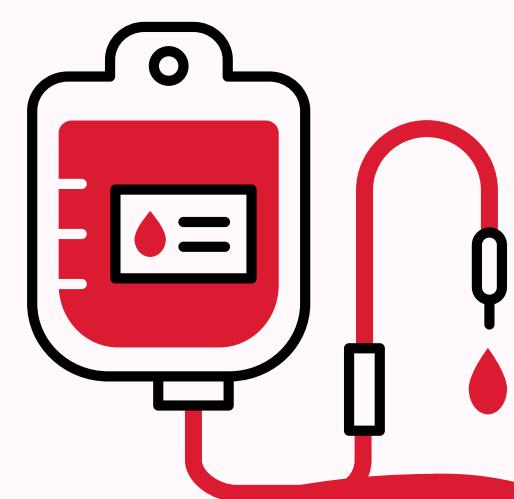
```
create database databaseproject;
use databaseproject ;
CREATE TABLE clinicalanalyst (
    id int(11) NOT NULL,
    name varchar(255) DEFAULT NULL,
    PRIMARY KEY (id)
);
```

```
CREATE TABLE manager (
    manager_id int(11) NOT NULL,
    name varchar(55) DEFAULT NULL,
    PRIMARY KEY (manager_id)
);
```

```
CREATE TABLE registrationteam (
    reg_id int(11) NOT NULL,
    name varchar(10) DEFAULT NULL,
    PRIMARY KEY (reg_id)
);
```

```
CREATE TABLE donor (
    d_id int(11) NOT NULL,
    name varchar(255) DEFAULT NULL,
    age int(11) DEFAULT NULL,
    gender varchar(10) DEFAULT NULL,
    bgroup varchar(20) DEFAULT NULL,
    disease varchar(20) DEFAULT NULL,
    bquantity varchar(20) DEFAULT NULL,
    d_date date DEFAULT NULL,
    PRIMARY KEY (d_id)
);
```

```
CREATE TABLE contact_donor
(id int(11) NOT NULL,
CONSTRAINT donor_ibfk_1
FOREIGN KEY (d_id)
REFERENCES donor (d_id),
address varchar (255) DEFAULT
NULL,
contact varchar (20) DEFAULT
NULL,
PRIMARY KEY (id));
```



```
CREATE TABLE donor_registrationteam (
    d_id int(11) NOT NULL,
```

```

reg_id int(11) NOT NULL,
CONSTRAINT donor_registrationteam_ibfk_1 FOREIGN KEY (d_id)
REFERENCES donor (d_id),
CONSTRAINT donor_registrationteam_ibfk_2 FOREIGN KEY (reg_id)
REFERENCES registrationteam (reg_id);

```

```

CREATE TABLE patient (
id int(11) NOT NULL,
name varchar(255) DEFAULT NULL,
age int(11) DEFAULT NULL,
gender varchar(10) DEFAULT NULL,
PRIMARY KEY (id)
);

```

```

CREATE TABLE contact_patient (id int(11) NOT NULL,
CONSTRAINT patient_ibfk_1
FOREIGN KEY (id_patient) REFERENCES patient (id) ,
address varchar(255) DEFAULT NULL, contact varchar(20) DEFAULT
NULL, PRIMARY KEY (id));

```

```

CREATE TABLE registrationteam_patient (
reg_id int(11) NOT NULL,
patient_id int(11) NOT NULL,
PRIMARY KEY (reg_id,patient_id),
KEY patient_id (patient_id),
CONSTRAINT registrationteam_patient_ibfk_1 FOREIGN KEY (reg_id)
REFERENCES registrationteam (reg_id),
CONSTRAINT registrationteam_patient_ibfk_2
FOREIGN KEY (patient_id) REFERENCES patient (id)
);

```

```

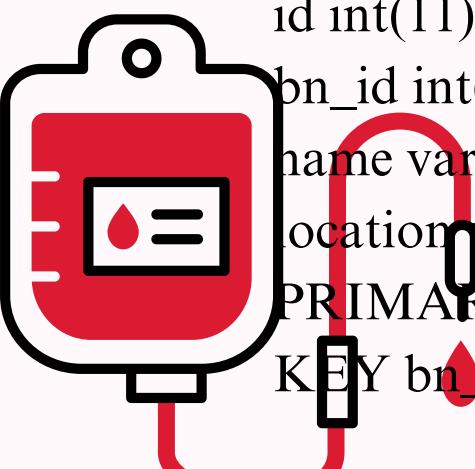
CREATE TABLE bloodbank (
bn_id int(11) NOT NULL,
manager_id int(11) DEFAULT NULL,
name varchar(55) DEFAULT NULL,
location varchar(255) DEFAULT NULL,
PRIMARY KEY (bn_id),
UNIQUE KEY manager_id (manager_id),
CONSTRAINT bloodbank_ibfk_1 FOREIGN KEY (manager_id)
REFERENCES manager (manager_id)
);

```

```

CREATE TABLE hospital (
id int(11) NOT NULL,
bn_id int(11) DEFAULT NULL,
name varchar(255) DEFAULT NULL,
location varchar(255) DEFAULT NULL,
PRIMARY KEY (id),
KEY bn_id (bn_id),

```



CONSTRAINT hospital\_ibfk\_1 FOREIGN KEY (bn\_id)

REFERENCES bloodbank (bn\_id)

);

CREATE TABLE blood (

bl\_id int(11) NOT NULL,

b\_group varchar(10) DEFAULT NULL,

d\_id int(11) DEFAULT NULL,

bn\_id int(11) DEFAULT NULL,

PRIMARY KEY (bl\_id) CONSTRAINT blood\_ibfk\_1,

FOREIGN KEY (d\_id) REFERENCES donor (d\_id),

CONSTRAINT blood\_ibfk\_2 FOREIGN KEY (bn\_id)

REFERENCES bloodbank (bn\_id),

PRIMARY KEY(bl\_id);

);

CREATE TABLE blood\_clinicalanalyst (

bl\_id int(11) NOT NULL,

analyst\_id int(11) NOT NULL,

CONSTRAINT blood\_clinicalanalyst\_ibfk\_1 FOREIGN KEY

(bl\_id)

REFERENCES blood (bl\_id),

CONSTRAINT blood\_clinicalanalyst\_ibfk\_2 FOREIGN KEY

(analyst\_id)

REFERENCES clinicalanalyst (id)

);

CREATE TABLE blood\_transfusion (

id int(11) NOT NULL AUTO\_INCREMENT,

patient\_id int(11) DEFAULT NULL,

blood\_id int(11) DEFAULT NULL,

quantity int(11) DEFAULT NULL,

intakeDate date DEFAULT NULL,

transfusion\_date date DEFAULT NULL,

PRIMARY KEY (id),

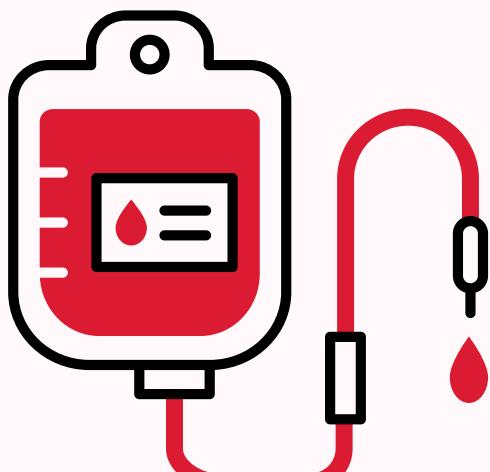
CONSTRAINT blood\_transfusion\_ibfk\_1 FOREIGN KEY

(patient\_id) REFERENCES patient (id),

CONSTRAINT blood\_transfusion\_ibfk\_2 FOREIGN KEY

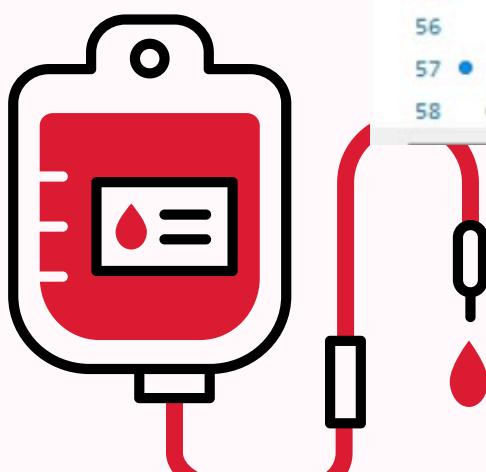
(blood\_id) REFERENCES blood (bl\_id)

);



# MYSQL CODE OF BLOOD BANK DATABASE CREATION SCREENSHOTS

```
1 •  create database databaseproject;
2   use databaseproject ;
3   CREATE TABLE clinicalanalyst (
4     id int(11) NOT NULL,
5     name varchar(255) DEFAULT NULL,
6     PRIMARY KEY (id)
7   );
8 •  CREATE TABLE manager (
9     manager_id int(11) NOT NULL,
10    name varchar(55) DEFAULT NULL,
11    PRIMARY KEY (manager_id)
12  );
13 •  CREATE TABLE registrationteam (
14    reg_id int(11) NOT NULL,
15    name varchar(10) DEFAULT NULL,
16    PRIMARY KEY (reg_id)
17  );
18 •  CREATE TABLE donor (
19    d_id int(11) NOT NULL,
20    name varchar(255) DEFAULT NULL,
21    age int(11) DEFAULT NULL,
22    gender varchar(10) DEFAULT NULL,
23    bgroup varchar(20) DEFAULT NULL,
24    disease varchar(20) DEFAULT NULL,
25    bquantity varchar(20) DEFAULT NULL,
26    d_date date DEFAULT NULL,
27    PRIMARY KEY (d_id)
28 • );CREATE TABLE contact_donor
29   (id int(11) NOT NULL, CONSTRAINT donor_ibfk_1
30   FOREIGN KEY (d_id)
31   REFERENCES donor (d_id),
32   );CREATE TABLE contact_donor
33   (id int(11) NOT NULL, CONSTRAINT donor_ibfk_1
34   FOREIGN KEY (d_id)
35   REFERENCES donor (d_id),
36   address varchar (255) DEFAULT
37   NULL,
38   phone varchar (20) DEFAULT
39   NULL,
40   PRIMARY KEY (id));
41 •  CREATE TABLE contact_patient
42   (id int(11) NOT NULL,
43   CONSTRAINT patient_ibfk_1
44   FOREIGN KEY (patient_id) REFERENCES patient (id) ,
45   address varchar(255) DEFAULT NULL, contact varchar(20) DEFAULT NULL, PRIMARY KEY (id));
46 •  CREATE TABLE donor_registrationteam (
47   d_id int(11) NOT NULL,
48   reg_id int(11) NOT NULL,
49   KEY reg_id (reg_id),
50   CONSTRAINT donor_registrationteam_ibfk_1 FOREIGN KEY (d_id)
51   REFERENCES donor (d_id),
52   CONSTRAINT donor_registrationteam_ibfk_2 FOREIGN KEY (reg_id)
53   REFERENCES registrationteam (reg_id)
54   );
55 •  CREATE TABLE patient (
56   id int(11) NOT NULL,
57   name varchar(255) DEFAULT NULL,
58   age int(11) DEFAULT NULL,
59   gender varchar(10) DEFAULT NULL,
60   PRIMARY KEY (id)
61 • );CREATE TABLE contact_patient
62   (id int(11) NOT NULL,
```

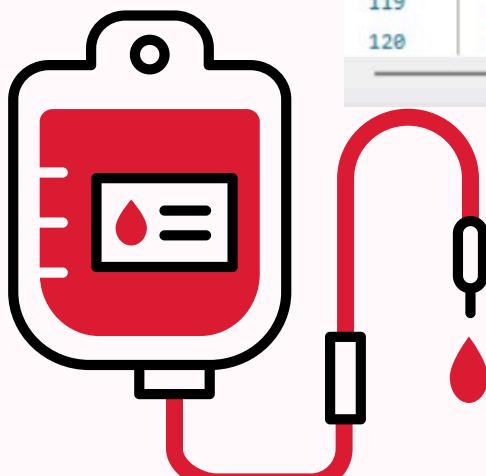


# MYSQL CODE OF BLOOD BANK DATABASE CREATION SCREENSHOTS

```

59   CONSTRAINT patient_ibfk_1
60   FOREIGN KEY (id_patient) REFERENCES patient (id) ,
61   address varchar(255) DEFAULT NULL, contact varchar(20) DEFAULT NULL, PRIMARY KEY (id));
62 • ○ CREATE TABLE registrationteam_patient (
63   reg_id int(11) NOT NULL,
64   patient_id int(11) NOT NULL,
65   PRIMARY KEY (reg_id,patient_id),
66   KEY patient_id (patient_id),
67   CONSTRAINT registrationteam_patient_ibfk_1 FOREIGN KEY (reg_id)
68   REFERENCES registrationteam (reg_id),
69   CONSTRAINT registrationteam_patient_ibfk_2
70   FOREIGN KEY (patient_id) REFERENCES patient (id)
71 );
72 • ○ CREATE TABLE bloodbank (
73   bn_id int(11) NOT NULL,
74   manager_id int(11) DEFAULT NULL,
75   name varchar(55) DEFAULT NULL,
76   location varchar(255) DEFAULT NULL,
77   PRIMARY KEY (bn_id),
78   UNIQUE KEY manager_id (manager_id),
79   CONSTRAINT bloodbank_ibfk_1 FOREIGN KEY (manager_id) REFERENCES manager (manager_id)
80 );
81 • ○ CREATE TABLE hospital (
82   id int(11) NOT NULL,
83   bn_id int(11) DEFAULT NULL,
84   name varchar(255) DEFAULT NULL,
85   location varchar(255) DEFAULT NULL,
86   PRIMARY KEY (id),
87   KEY bn_id (bn_id),
88   CONSTRAINT hospital_ibfk_1 FOREIGN KEY (bn_id) REFERENCES bloodbank (bn_id)
89 );
90 • ○ CREATE TABLE blood (
91   bl_id int(11) NOT NULL,
92   b_group varchar(10) DEFAULT NULL,
93   d_id int(11) DEFAULT NULL,
94   bn_id int(11) DEFAULT NULL,
95   PRIMARY KEY (bl_id),
96   KEY d_id (d_id),
97   KEY bn_id (bn_id),
98   CONSTRAINT blood_ibfk_1 FOREIGN KEY (d_id) REFERENCES donor (d_id),
99   CONSTRAINT blood_ibfk_2 FOREIGN KEY (bn_id) REFERENCES bloodbank (bn_id)
100 );
101 • ○ CREATE TABLE blood_clinicalanalyst (
102   bl_id int(11) NOT NULL,
103   analyst_id int(11) NOT NULL,
104   KEY analyst_id (analyst_id),
105   CONSTRAINT blood_clinicalanalyst_ibfk_1 FOREIGN KEY (bl_id)
106   REFERENCES blood (bl_id),
107   CONSTRAINT blood_clinicalanalyst_ibfk_2 FOREIGN KEY (analyst_id)
108   REFERENCES clinicalanalyst (id)
109 );
110 • ○ CREATE TABLE blood_transfusion (
111   id int(11) NOT NULL AUTO_INCREMENT,
112   patient_id int(11) DEFAULT NULL,
113   blood_id int(11) DEFAULT NULL,
114   quantity int(11) DEFAULT NULL,
115   intakeDate date DEFAULT NULL,
116   transfusion_date date DEFAULT NULL,
117   PRIMARY KEY (id),
118   KEY patient_id (patient_id),
119   KEY blood_id (blood_id),
120   CONSTRAINT blood_transfusion_ibfk_1 FOREIGN KEY (patient_id) REFERENCES patient (id),

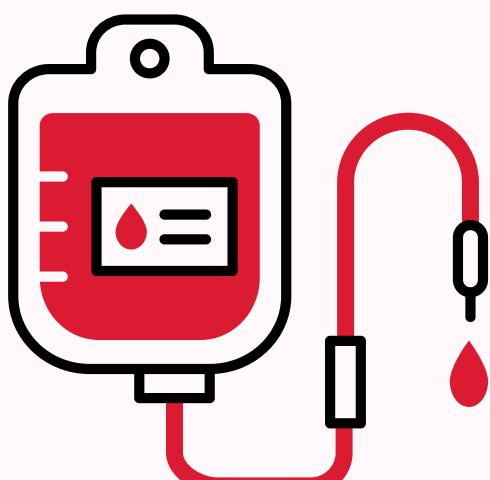
```



# OUR TABLES IN DATABASE

- Table:clinicalanalyst

	id	name
▶	1	Weel Smeth
	2	Jolen
	3	Tom
	4	Kim lee
	5	Sela
	6	slena
	7	Chan
	8	Fleix
	9	Tolean
	10	jeemry
●	NULl	NULl



- Table:Donor

SQL File 1 donor donor

```
1 • SELECT * FROM databaseproject.donor;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

d_id	name	age	gender	bgroup	disease	bquantity	d_date
1	Alex	25	Male	A+	Scabies	500 ml	2023-01-01
2	Hala	30	Female	B+	Scarlet fever	400 ml	2023-02-01
3	David	35	Male	B-	Schizophrenia	350 ml	2024-03-01
4	Mila	40	Female	AB+	Scoliosis	450 ml	2023-04-01
5	Sam	34	Male	AB-	Shingles	550 ml	2024-05-01
6	Kyle	28	Female	B+	Shortness of breath	300 ml	2023-06-01
7	Erik	37	Male	O+	Schizophrenia	400ml	2024-03-01
8	Lina	26	Female	O-	Shingles	250ml	2023-10-10
9	Adam	42	Male	A+	Scoliosis	450ml	2023-02-09
10	Alexa	32	Female	AB-	Shortness of breath	350ml	2023-12-24
*	HULL	HULL	HULL	HULL	HULL	HULL	HULL

donor 1 x Apply

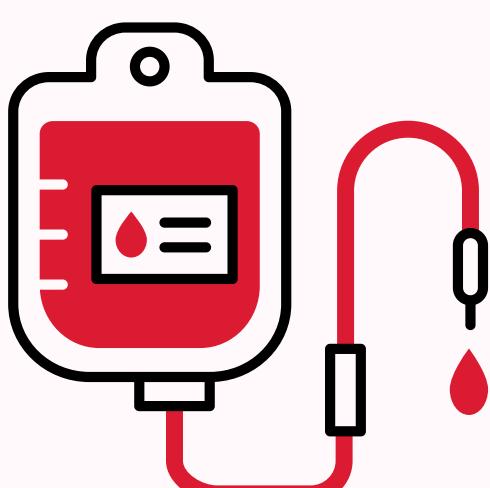
- Table:contact\_Donor

SQL File 1 contact\_donor

```
1 • SELECT * FROM databaseproject.contact_donor;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

id	d_id	address	phone
1	1	apt. 5A,usa	0555339200
2	2	Suite 102,spain	0548299896
3	3	43 Rue Jouffroy	0544869022
4	4	4-1 Kio, Japan	0592154766
5	5	Floor, Australia	0508053367
6	6	Suite 102,usa	0563866533
7	7	Vinbaletet 34,spain	0500286441
8	8	Bronok,Singapore	0598003786
9	9	Spinnaker,usa	0548669499
10	10	Estrada Portugal	0580666030
*	HULL	HULL	HULL

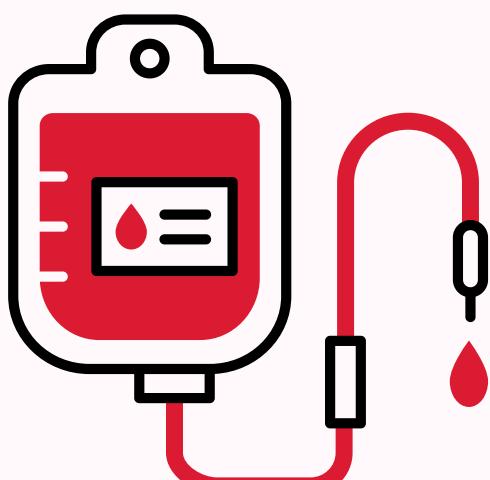


- Table:patient

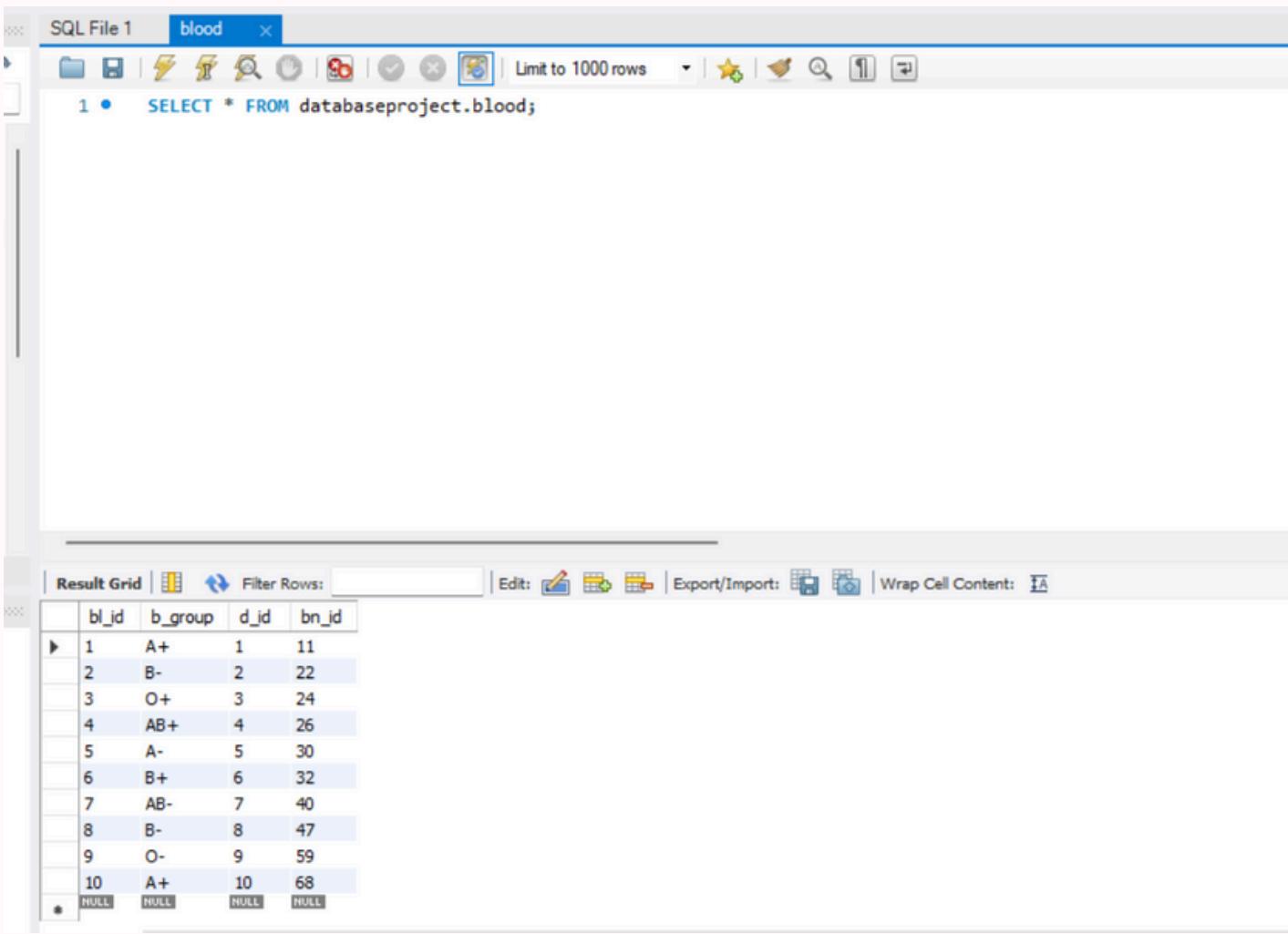
	<b>id</b>	<b>name</b>	<b>age</b>	<b>gender</b>
▶	1	Jhon Beech	35	Male
	2	Ann Ford	42	Male
	3	Lana Waxe	28	Female
	4	Alex Tles	50	Male
	5	Cristiano Tles	37	Male
	6	Diana Lee	39	Female
	7	Sara Smeth	20	Female
	8	Kai Lee	12	Male
	9	Talisca	33	Male
	10	Sara weel	18	Female

- Table:contact\_patient

	<b>id</b>	<b>id_patient</b>	<b>name</b>	<b>address</b>	<b>contact</b>
▶	1	1	Jhon Beech	apt. 5A,usa	40.32.2555
	2	2	Ann Ford	54,rue Royale	7025551838
	3	3	Lana Waxe	Vinbaeltet 34,spain	95204555
	4	4	Alex Tles	8489 Strong St.	9555 98-07
	5	5	Cristiano Tles	149 Spinnaker,usa	4155551450
	6	6	Diana Lee	897 Long Airport Avenue	25542-7555
	7	7	Sara Smeth	67, rue des Cinquante Otages	49 6 6690
	8	8	Kai Lee	Lyonerstr. 34	6505555787
	9	9	Talisca	5557 North Pendale Street	2125557818
	10	10	Sara weel	Erling Skakkes gate 78	2035552570



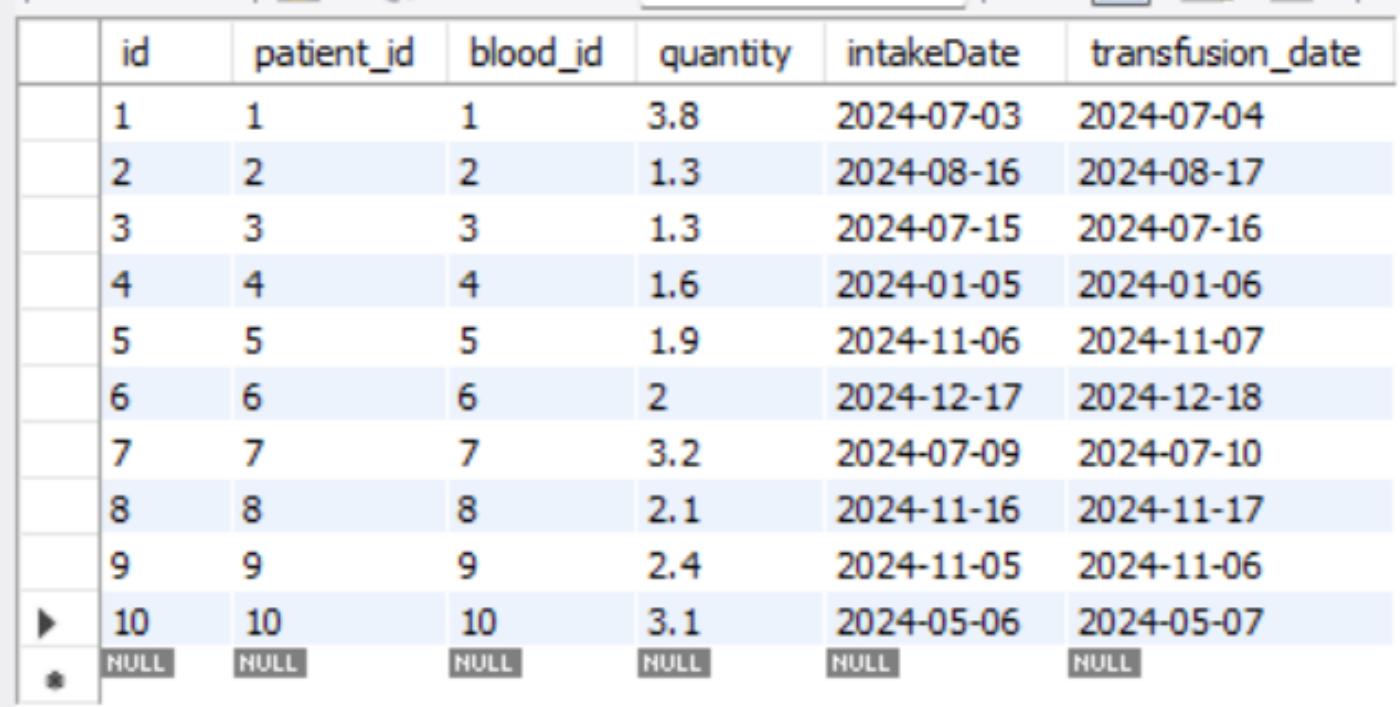
- Table:blood



The screenshot shows the MySQL Workbench interface with a query editor window titled "SQL File 1" containing the SQL command: "SELECT \* FROM databaseproject.blood;". Below the query is a "Result Grid" showing the data from the "blood" table. The table has four columns: bl\_id, b\_group, d\_id, and bn\_id. The data consists of 10 rows, each representing a blood group and its corresponding ID values.

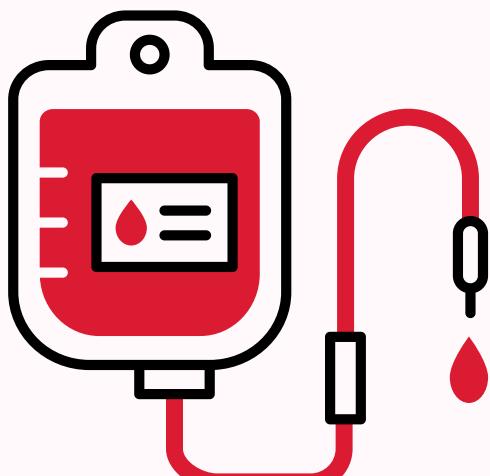
bl_id	b_group	d_id	bn_id
1	A+	1	11
2	B-	2	22
3	O+	3	24
4	AB+	4	26
5	A-	5	30
6	B+	6	32
7	AB-	7	40
8	B-	8	47
9	O-	9	59
10	A+	10	68
*	HULL	HULL	HULL

- Table: Blood\_transfusion



The screenshot shows the MySQL Workbench interface with a query editor window containing the SQL command: "SELECT \* FROM databaseproject.Blood\_transfusion;". Below the query is a "Result Grid" showing the data from the "Blood\_transfusion" table. The table has six columns: id, patient\_id, blood\_id, quantity, intakeDate, and transfusion\_date. The data consists of 10 rows, each representing a transfusion record with its details.

id	patient_id	blood_id	quantity	intakeDate	transfusion_date
1	1	1	3.8	2024-07-03	2024-07-04
2	2	2	1.3	2024-08-16	2024-08-17
3	3	3	1.3	2024-07-15	2024-07-16
4	4	4	1.6	2024-01-05	2024-01-06
5	5	5	1.9	2024-11-06	2024-11-07
6	6	6	2	2024-12-17	2024-12-18
7	7	7	3.2	2024-07-09	2024-07-10
8	8	8	2.1	2024-11-16	2024-11-17
9	9	9	2.4	2024-11-05	2024-11-06
10	10	10	3.1	2024-05-06	2024-05-07
*	NULL	NULL	NULL	NULL	NULL

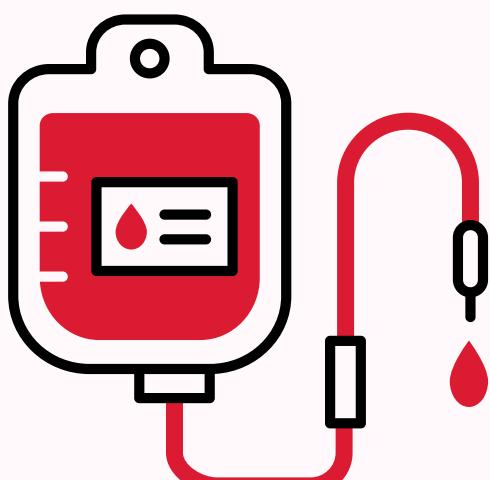


- Table: Manager

	manager_id	name
▶	1	Ali
	2	Fahad
	3	Ahmed
	4	Yaser
	5	Saud
	6	Saad
	7	Rayan
	8	Jaser
	9	Fahad
	10	Mohammed
	NULL	NULL

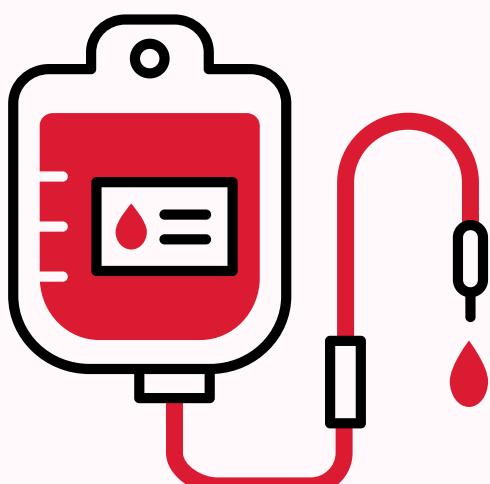
- Table: registrationteam

	reg_id	name
▶	1	Faris
	2	Hassan
	3	Sami
	4	Khalead
	5	Sultan
	6	Badr
	7	Rayan
	8	Saad
	9	Fhaad
	10	Rakan
	NULL	NULL



- Table: registrationteam\_Patinet

	reg_id	patient_id
▶	1	1
	2	2
	3	3
	4	4
	5	5
	6	6
	7	7
	8	8
	9	9
	10	10
	HULL	HULL

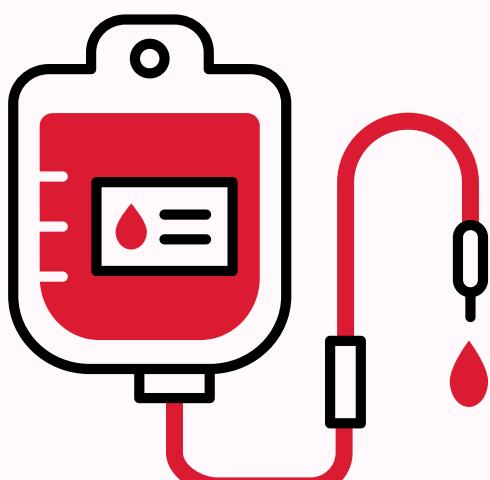


- Table: donor\_registrationteam

	d_id	reg_id
▶	1	1
	2	2
	3	3
	4	4
	5	5
	6	6
	7	7
	8	8
	9	9
	10	10

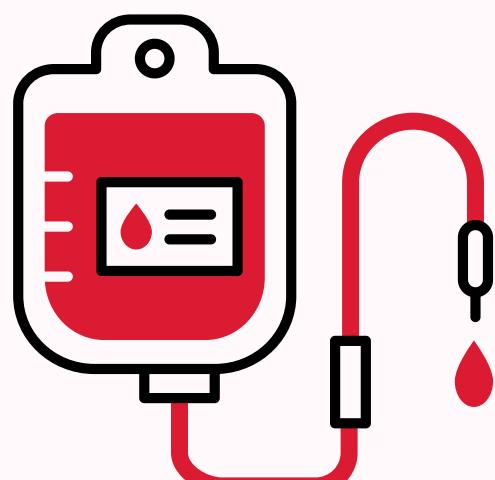
- Table: blood\_clinicalanalyst

	bl_id	analyst_id
▶	1	1
	2	2
	3	3
	4	4
	5	5
	6	6
	7	7
	8	8
	9	9
	10	10
	NUL	NUL



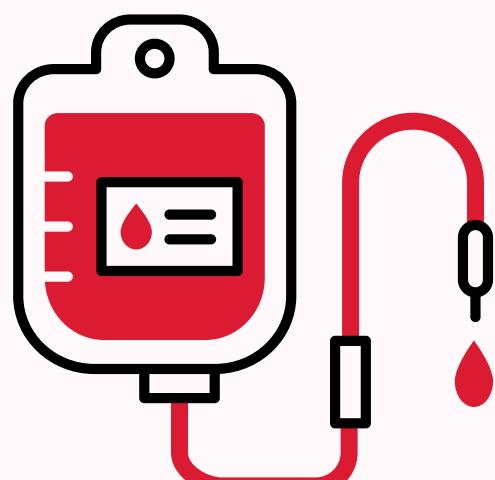
- Table:hospital

	<a href="#">id</a>	<a href="#">bn_id</a>	<a href="#">name</a>	<a href="#">location</a>
▶	1	30	City Hospital	New York
	2	22	Mercy Medical	Los Angeles
	3	32	Central Clinic	Chicago
	4	11	Unity Hospital	Chicago
	5	47	Unity Hospital	Miami
	6	24	Sunrise Medical	Boston
	7	59	River Health	Seattle
	8	26	Valley Care	Phoenix
	9	40	Summit Clinic	Denver
	10	68	Ocean Hospital	San Francisco

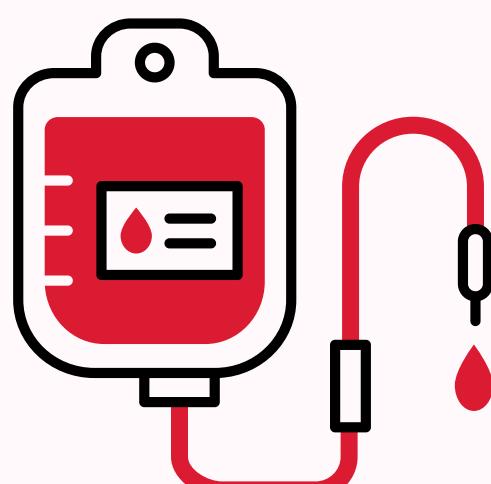
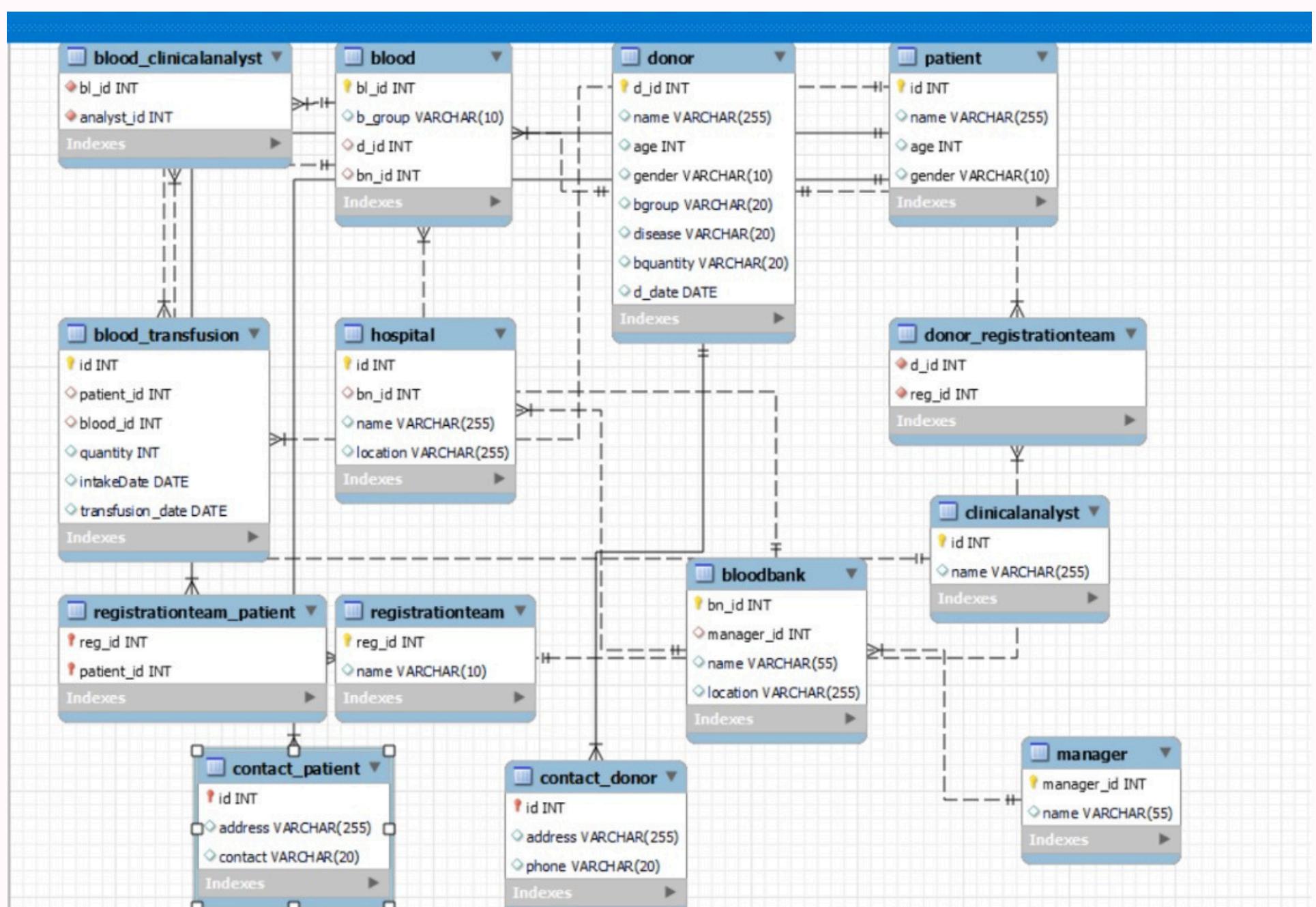


- Table:bloodbank

	bn_id	manager_id	name	location
▶	11	4	Unity Hospital	Houston
	22	2	Mercy Medical	Los Angeles
	24	6	Sunrise Clinic	San Francisco
	26	8	Harbor Clinic	Seattle
	30	1	City Hospital	New York
	32	3	Central Clinic	Chicago
	40	9	Summit Hospital	Denver
	47	5	Hope Clinic	Miami
	59	7	Valley Hospital	Phoenix
	68	10	Lakeside Clinic	Minneapolis



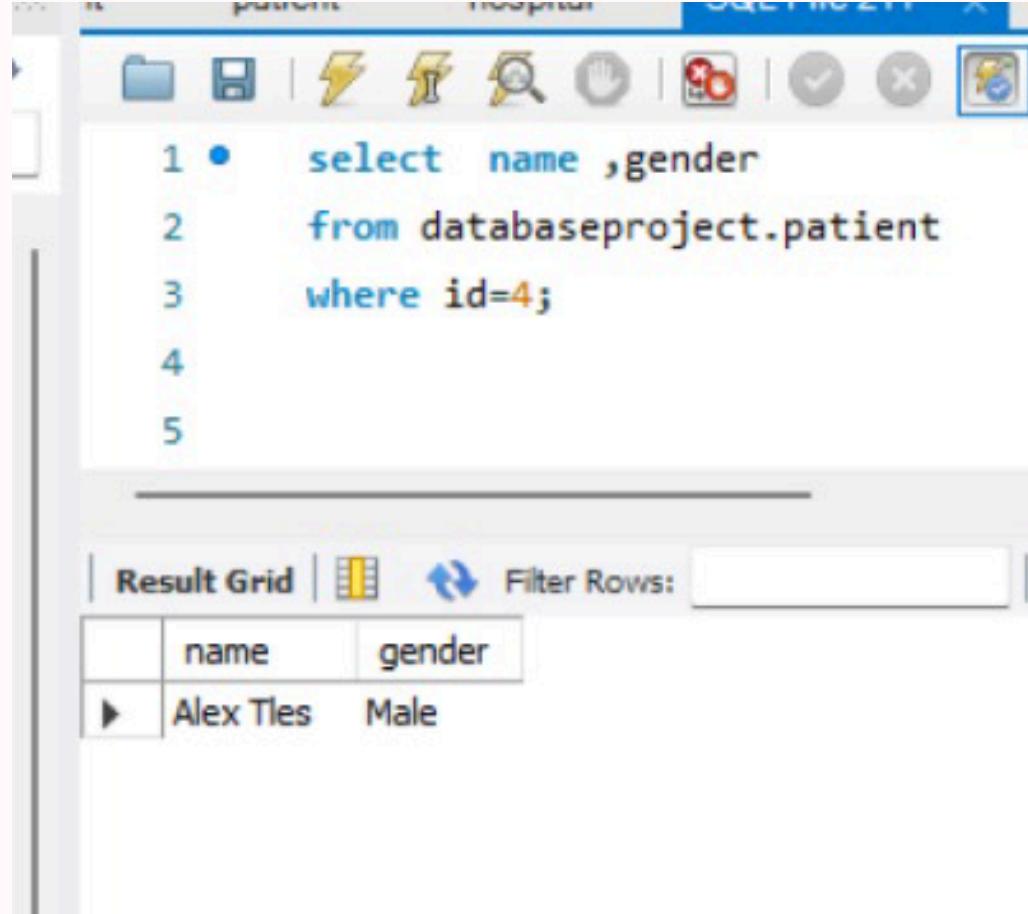
# DIAGRAM



# QUERYS

## 1. Patient table:

- Return the gender and name of patient with id 4



```
1 • select name ,gender
2   from databaseproject.patient
3   where id=4;
4
5
```

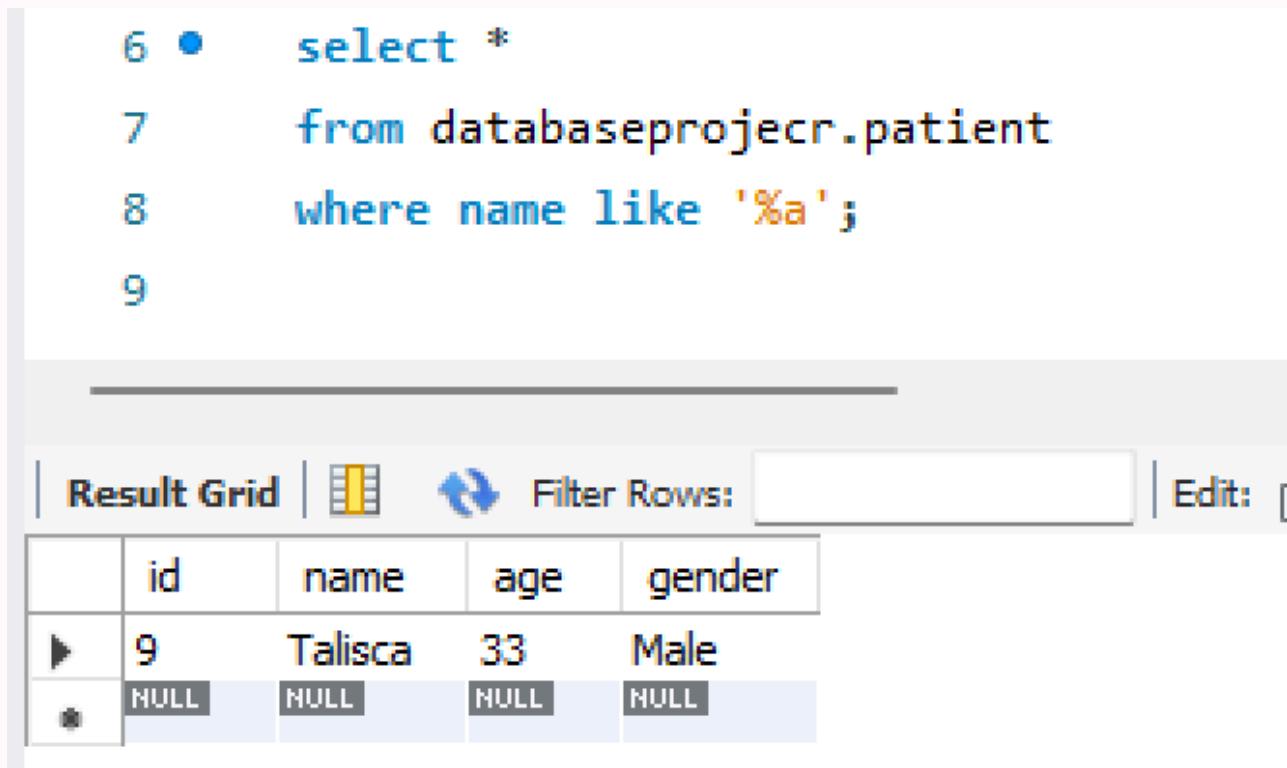
The screenshot shows a SQL query window with the following code:

```
1 • select name ,gender
2   from databaseproject.patient
3   where id=4;
4
5
```

Below the code is a result grid showing one row of data:

	name	gender
▶	Alex Ties	Male

- Return the patient whose Name End with character 'a'



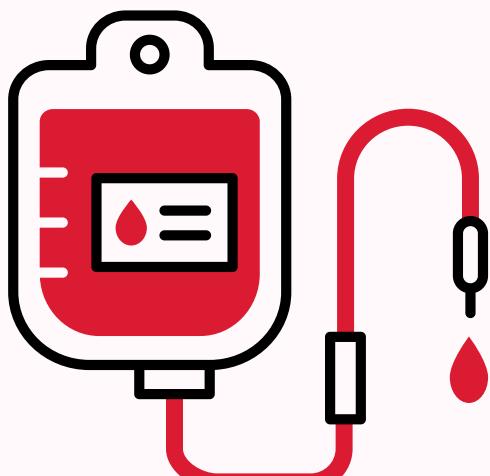
```
6 • select *
7   from databaseprojecr.patient
8   where name like '%a';
9
```

The screenshot shows a SQL query window with the following code:

```
6 • select *
7   from databaseprojecr.patient
8   where name like '%a';
9
```

Below the code is a result grid showing two rows of data:

	id	name	age	gender
▶	9	Talisca	33	Male
*	NULL	NULL	NULL	NULL



By Almaha

# QUERYS

## 2.Contact\_Patient table:

- Return the name and address and contact of patient and name orderd desc

```

8
9 •   select name ,address,contact
10    from databaseproject.contact_patient
11    order by name desc;
12
13
14
15
16
17
  
```

Result Grid | Filter Rows: Export: 

	name	address	contact
▶	Talisca	5557 North Pendale Street	2125557818
	Sara weel	Erling Skakkes gate 78	2035552570
	Sara Smeth	67, rue des Cinquante Otages	49 6 6690
	Lana Waxe	Vinbaeltet 34,spain	95204555
	Kai Lee	Lyonerstr. 34	6505555787
	Jhon Beech	apt. 5A,usa	40.32.2555
	Diana Lee	897 Long Airport Avenue	25542-7555
	Cristiano Tles	149 Spinnaker,usa	4155551450
	Ann Ford	54,rue Royale	7025551838
	Alex Tles	8489 Strong St.	9555 98-07

contact patient 5 x

- Display the name and id in single column not single set with new column name

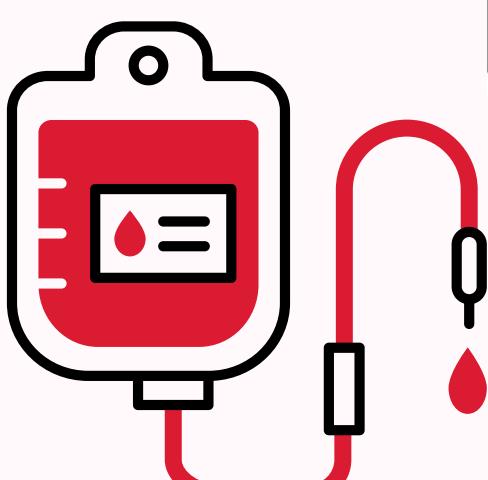
patient hospital SQL File 211\* x contact\_patient patient - Table

File Edit View Insert Tools Options Help

16
 17 • select concat(id , ' ' ,name ) As iD\_and\_name
 18 from databaseproject.contact\_patient
 19
 20

Result Grid | Filter Rows: Export:  Wrap Cell Content

	iD_and_name
▶	4 Alex Tles
	2 Ann Ford
	5 Cristiano Tles
	6 Diana Lee
	1 Jhon Beech
	8 Kai Lee
	3 Lana Waxe
	7 Sara Smeth
	10 Sara weel
	9 Talisca



By Almaha

# QUERYS

- Suppose you want to combine data from the donor and contact\_donor tables into a single result set:

SQL File 1\*

Result Grid	
d_id	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

Result 2 x

```

1 •  select d_id
2   FROM databaseproject.donor
3   UNION
4   SELECT d_id
5   FROM databaseproject.contact_donor;
  
```

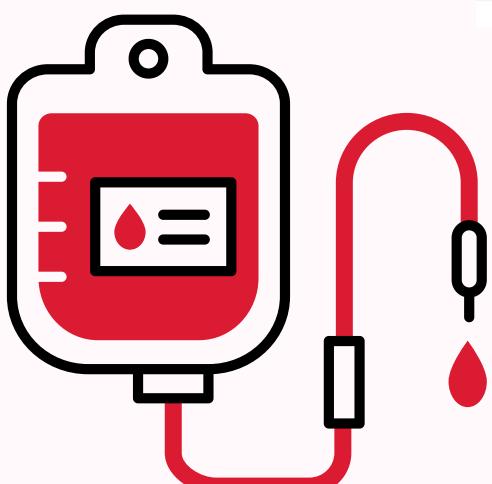
- Find all donor whose age is greater than 25:

SQL File 1\*

Result Grid	
name	age
Hala	30
David	35
Mila	40
Sam	34
Kyle	28
Erik	37
Lina	26
Adam	42
Alexa	32

```

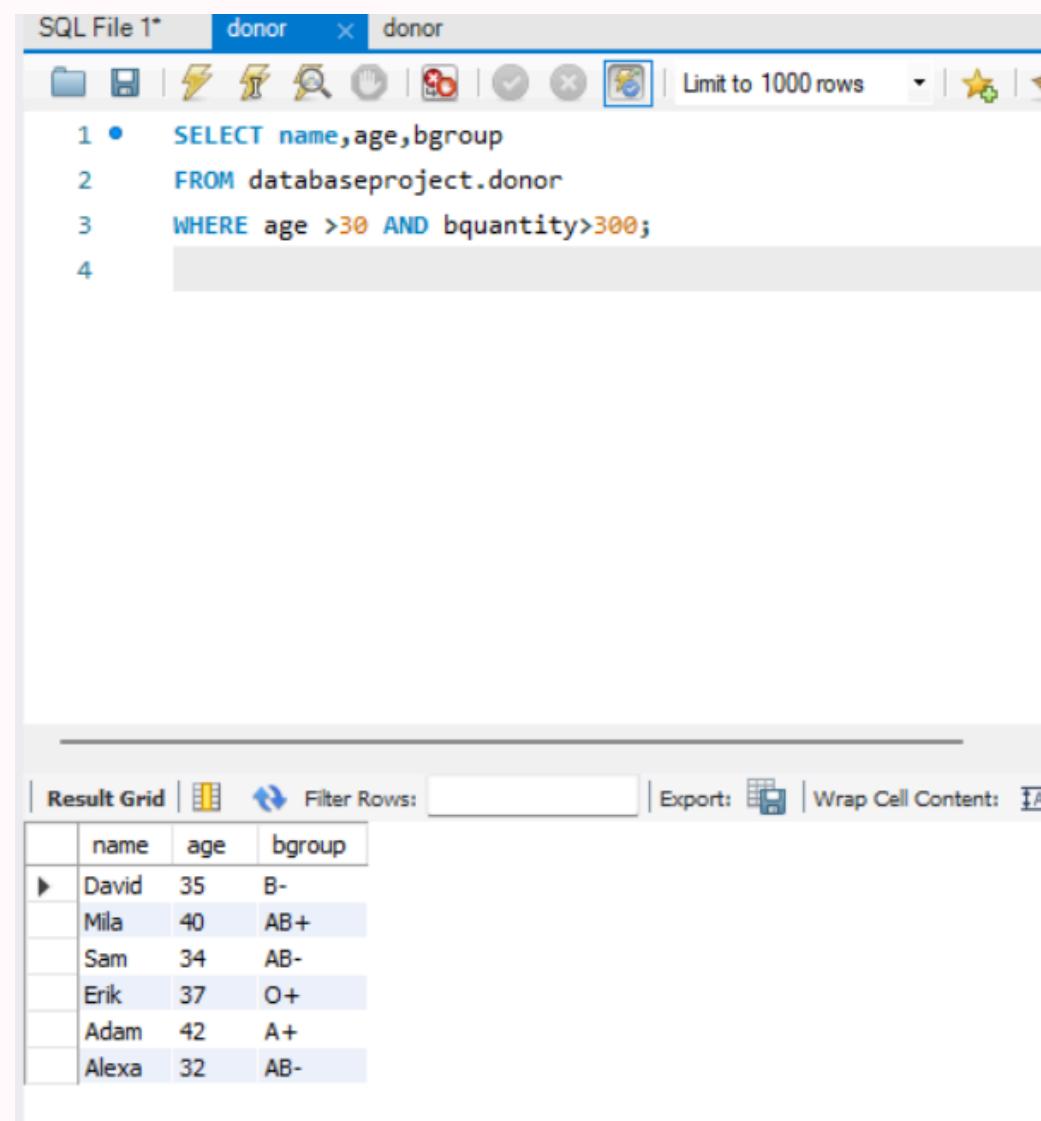
1 •  SELECT name,age
2   FROM databaseproject.donor
3   WHERE age > 25;
  
```



By Fahdah

# QUERYS

- Find all donor whose age is greater than 30 and bquantity greater than 300:



The screenshot shows a MySQL Workbench interface. The SQL editor tab contains the following query:

```

1 •  SELECT name,age,bgroup
2   FROM databaseproject.donor
3  WHERE age >30 AND bquantity>300;
4
  
```

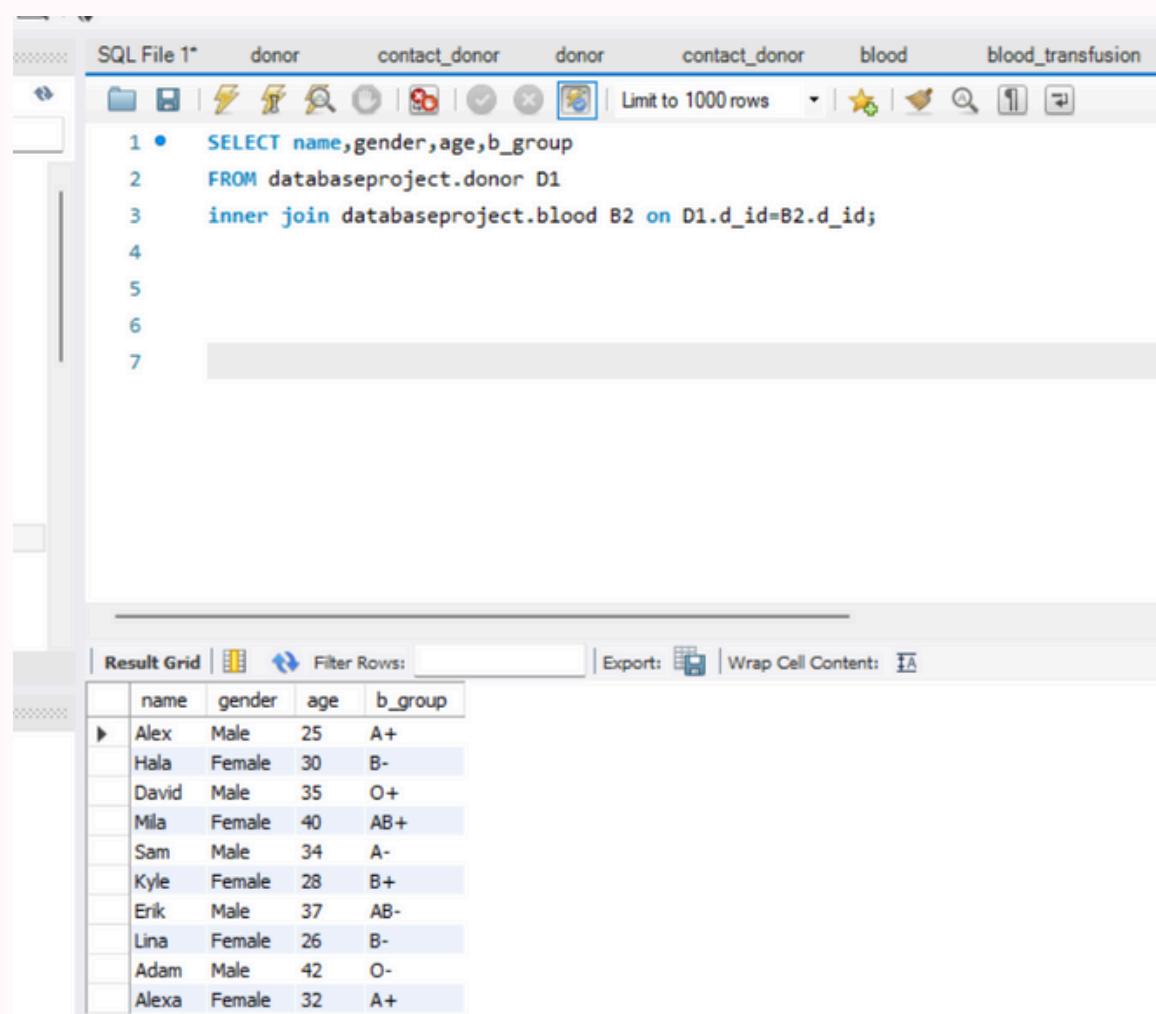
The results grid displays the following data:

	name	age	bgroup
▶	David	35	B-
	Mila	40	AB+
	Sam	34	AB-
	Erik	37	O+
	Adam	42	A+
	Alexa	32	AB-

- Two tables:

donor and blood:

Now, if you want to get:-- The name,gender and age from the donor table. The b\_group the blood table.



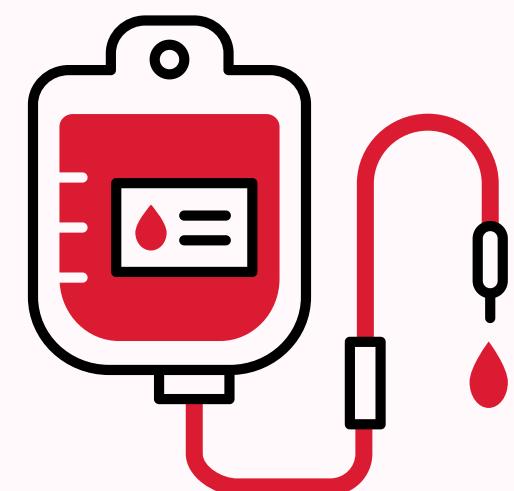
The screenshot shows a MySQL Workbench interface. The SQL editor tab contains the following query:

```

1 •  SELECT name,gender,age,b_group
2   FROM databaseproject.donor D1
3  inner join databaseproject.blood B2 on D1.d_id=B2.d_id;
4
5
6
7
  
```

The results grid displays the following data:

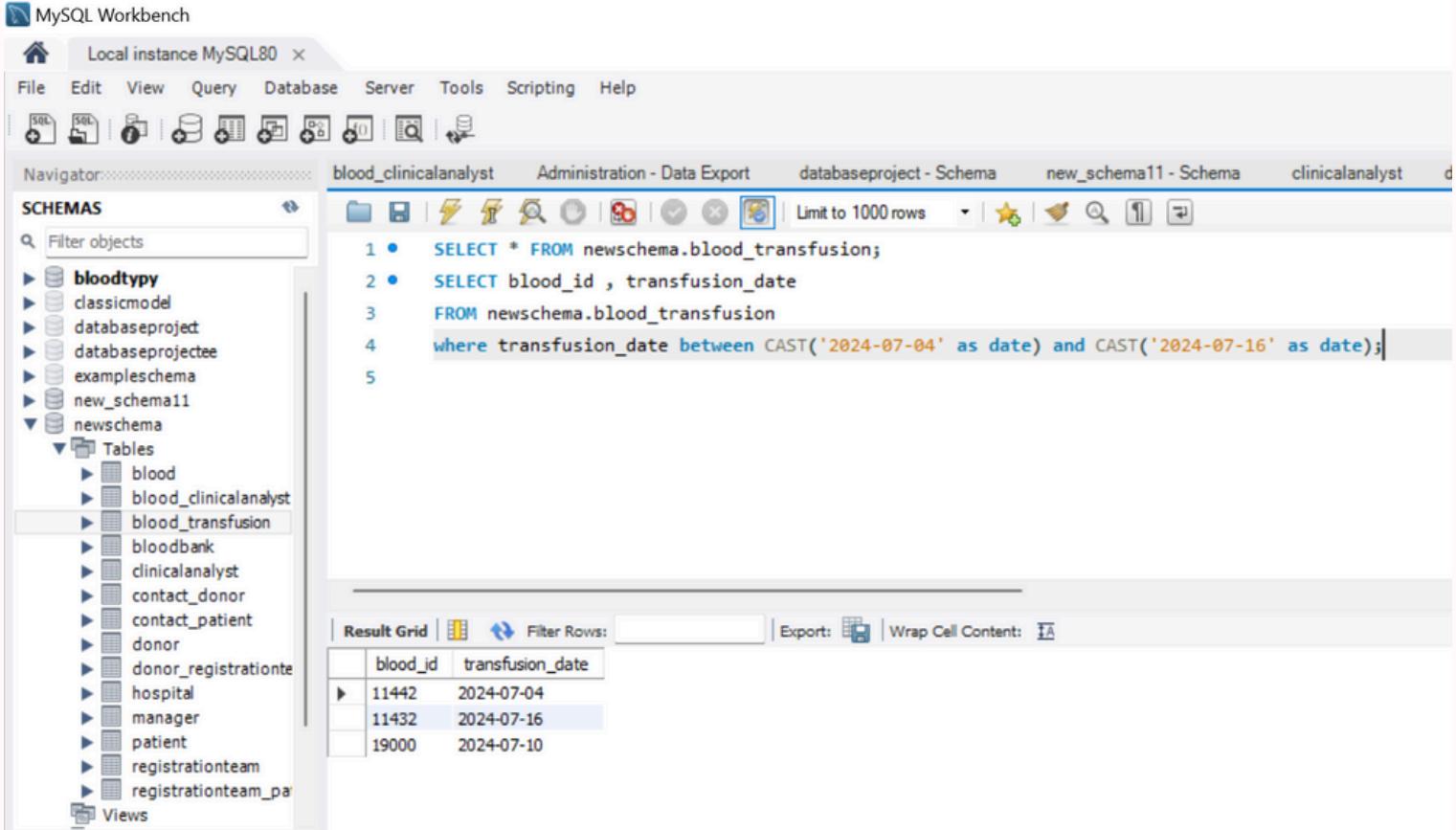
	name	gender	age	b_group
▶	Alex	Male	25	A+
	Hala	Female	30	B-
	David	Male	35	O+
	Mila	Female	40	AB+
	Sam	Male	34	A-
	Kyle	Female	28	B+
	Erik	Male	37	AB-
	Lina	Female	26	B-
	Adam	Male	42	O-
	Alexa	Female	32	A+



By Fahdah

# QUERYS

- Find the blood transfusions whose required date is 2024-07-04 to 2024-07-16:



```

MySQL Workbench
Local instance MySQL80 x
File Edit View Query Database Server Tools Scripting Help
Navigator: blood_clinicalanalyst Administration - Data Export databaseproject - Schema new_schema11 - Schema clinicalanalyst d
SCHEMAS
  Filter objects
  bloodtypy
  classicmodel
  databaseproject
  databaseprojectee
  exampleschema
  new_schema11
  newschema
    Tables
      blood
      blood_clinicalanalyst
      blood_transfusion
      bloodbank
      clinicalanalyst
      contact_donor
      contact_patient
      donor
      donor_registration
      hospital
      manager
      patient
      registrationteam
      registrationteam_pa
    Views
  
```

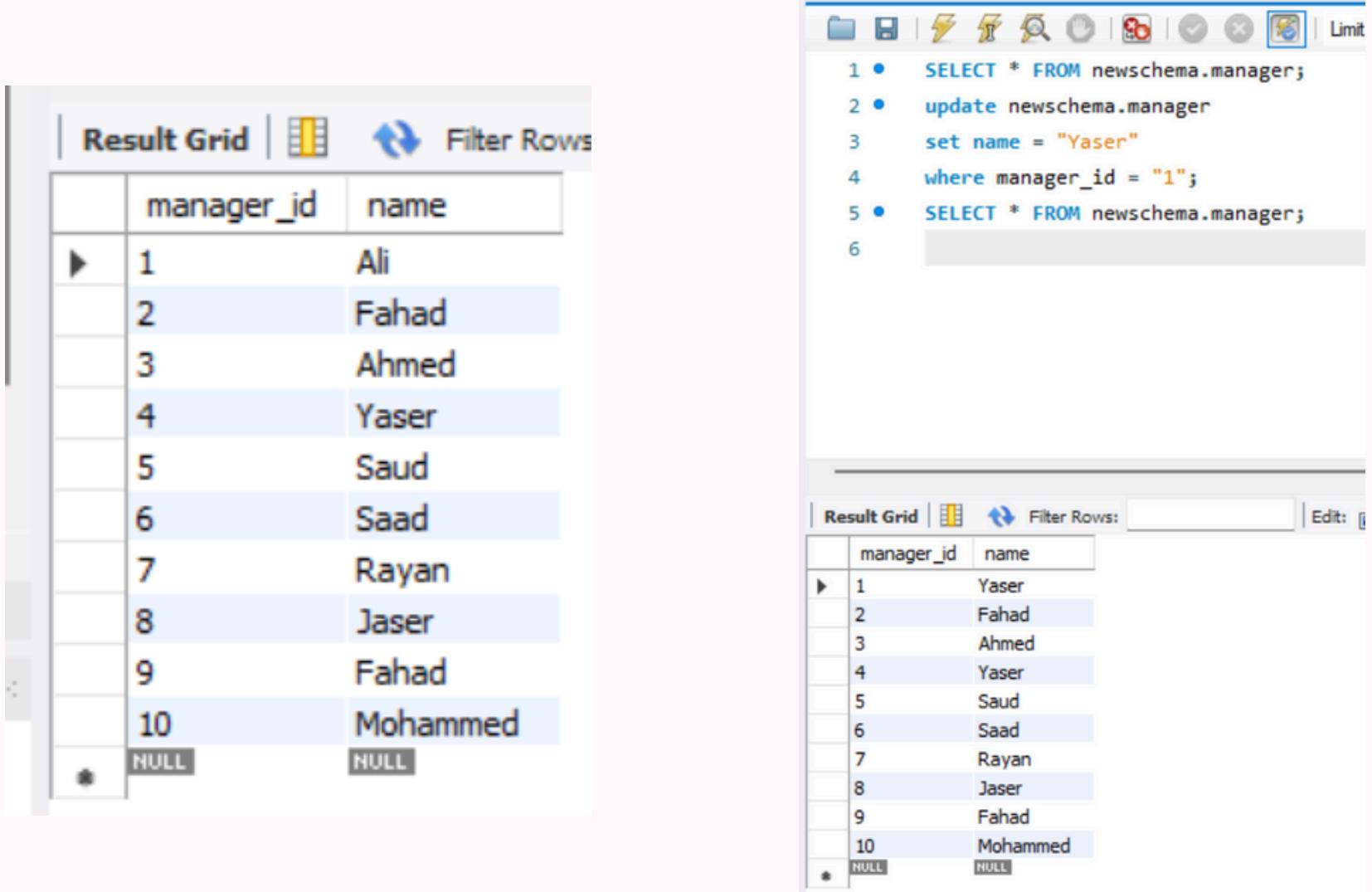
```

1 •  SELECT * FROM newschema.blood_transfusion;
2 •  SELECT blood_id , transfusion_date
3   FROM newschema.blood_transfusion
4   where transfusion_date between CAST('2024-07-04' as date) and CAST('2024-07-16' as date);
5

```

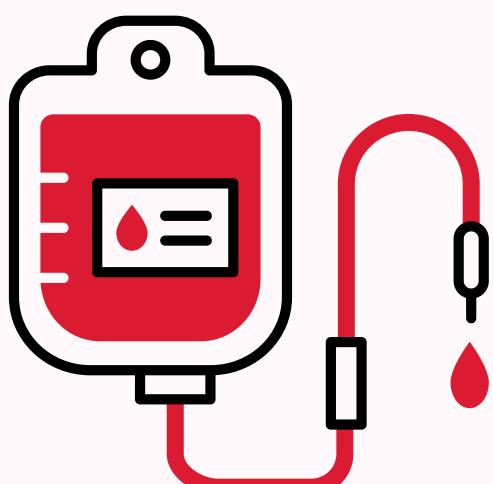
	blood_id	transfusion_date
▶	11442	2024-07-04
	11432	2024-07-16
	19000	2024-07-10

- Update the current name of the manager to the ID number of 1 to the name “Yaser”:



	manager_id	name
▶	1	Ali
	2	Fahad
	3	Ahmed
	4	Yaser
	5	Saud
	6	Saad
	7	Rayan
	8	Jaser
	9	Fahad
	10	Mohammed
*	NULL	NULL

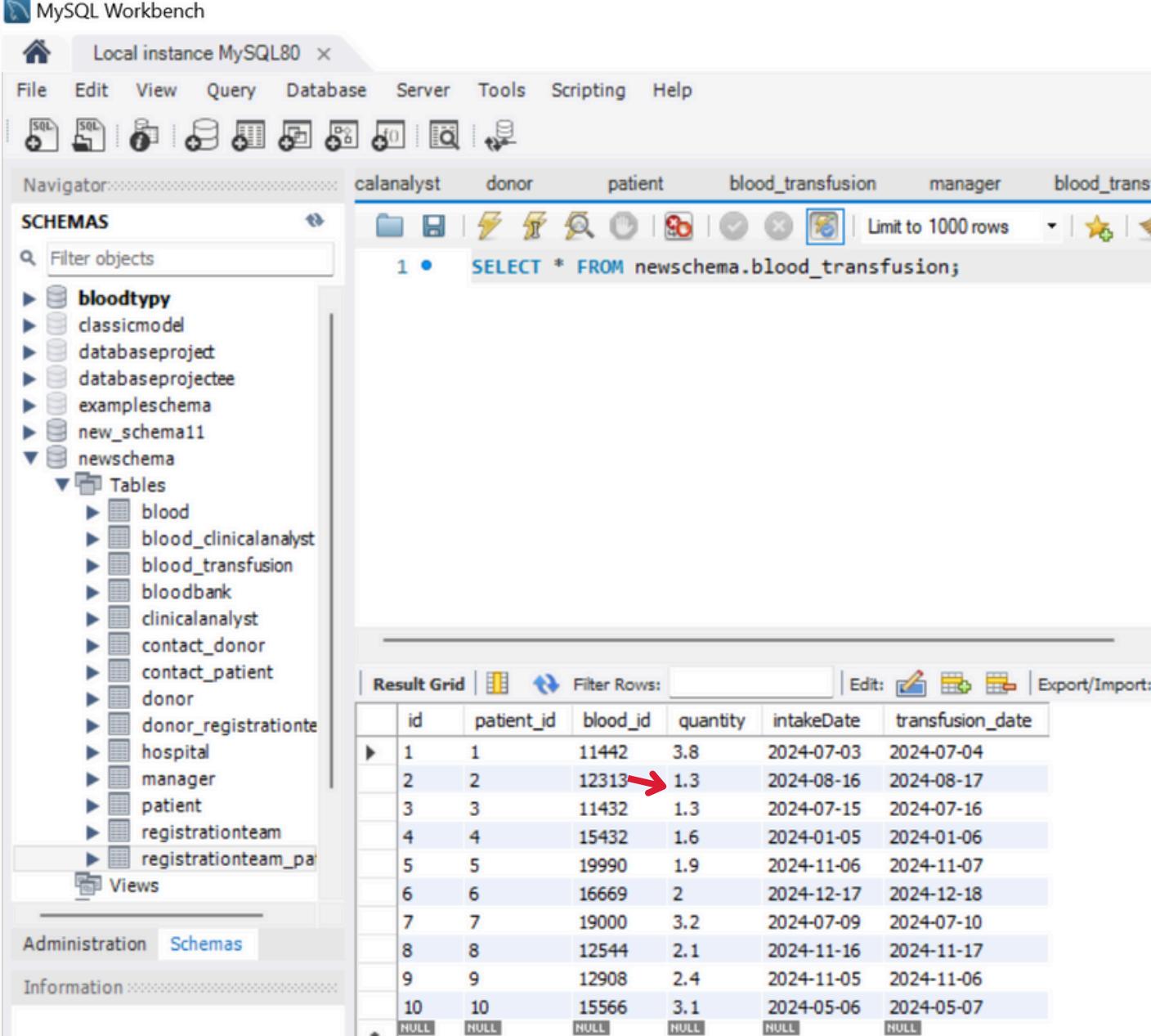
	manager_id	name
▶	1	Yaser
	2	Fahad
	3	Ahmed
	4	Yaser
	5	Saud
	6	Saad
	7	Rayan
	8	Jaser
	9	Fahad
	10	Mohammed
*	NULL	NULL



By Reem

# QUERYS

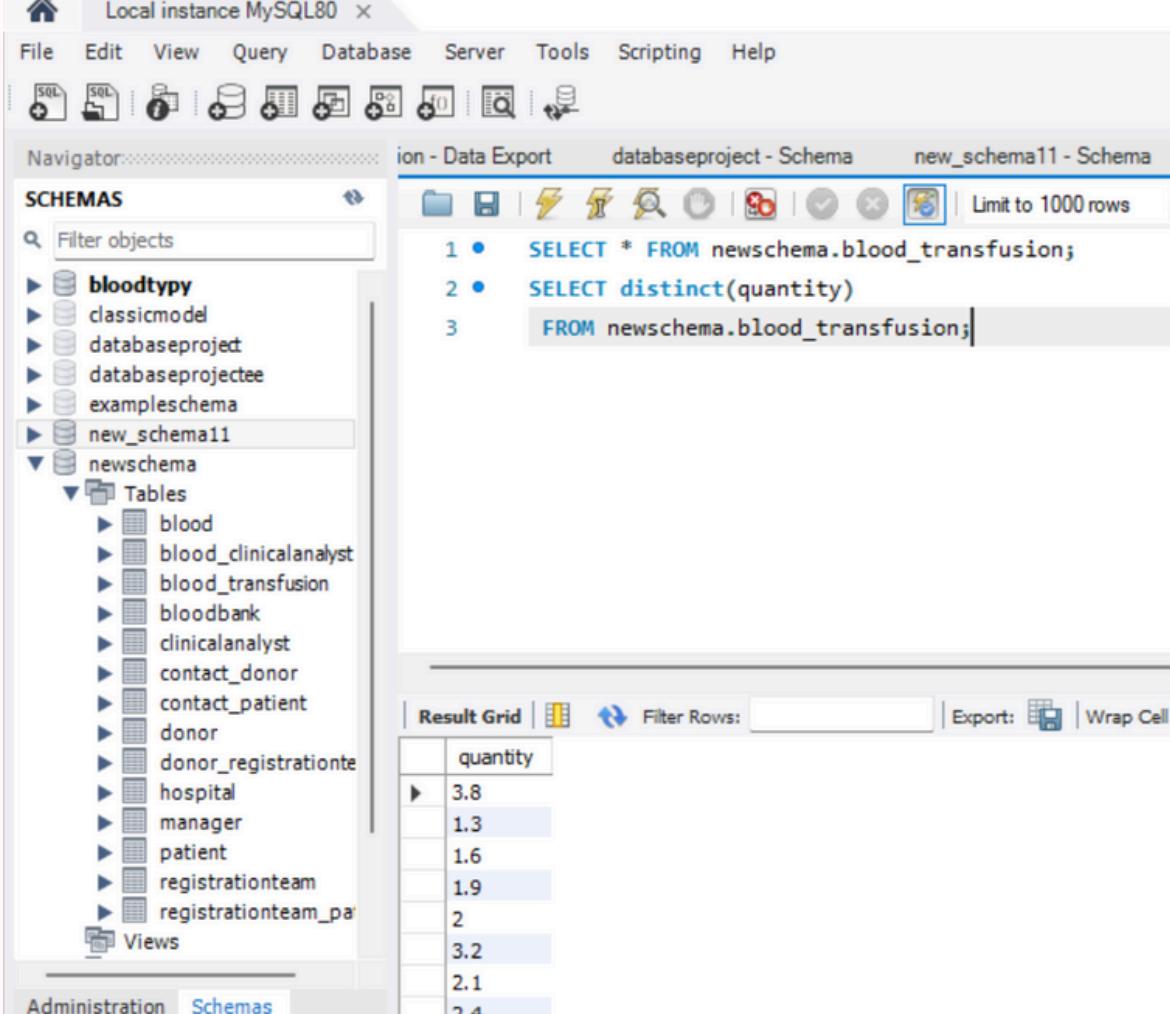
- Find a column in a table that has duplicate attributes:



The screenshot shows the MySQL Workbench interface. In the Navigator pane, under the 'newschema' schema, the 'Tables' section is expanded, showing various tables like blood, blood\_transfusion, etc. In the main SQL editor window, the query `SELECT * FROM newschema.blood_transfusion;` is run. The Result Grid displays 10 rows of data. The 'quantity' column contains several duplicate values: 3.8, 1.3, 1.3, 1.3, 1.6, 1.9, 2, 3.2, 2.1, and 2.4. The value '1.3' appears twice in the list.

	id	patient_id	blood_id	quantity	intakeDate	transfusion_date
1	1		11442	3.8	2024-07-03	2024-07-04
2	2		12313	1.3	2024-08-16	2024-08-17
3	3		11432	1.3	2024-07-15	2024-07-16
4	4		15432	1.6	2024-01-05	2024-01-06
5	5		19990	1.9	2024-11-06	2024-11-07
6	6		16669	2	2024-12-17	2024-12-18
7	7		19000	3.2	2024-07-09	2024-07-10
8	8		12544	2.1	2024-11-16	2024-11-17
9	9		12908	2.4	2024-11-05	2024-11-06
10	10		15566	3.1	2024-05-06	2024-05-07
*				NUL	NUL	NUL

- Use the appropriate query to remove the duplicate:



The screenshot shows the MySQL Workbench interface. In the Navigator pane, under the 'newschema' schema, the 'Tables' section is expanded. In the main SQL editor window, the following query is run:

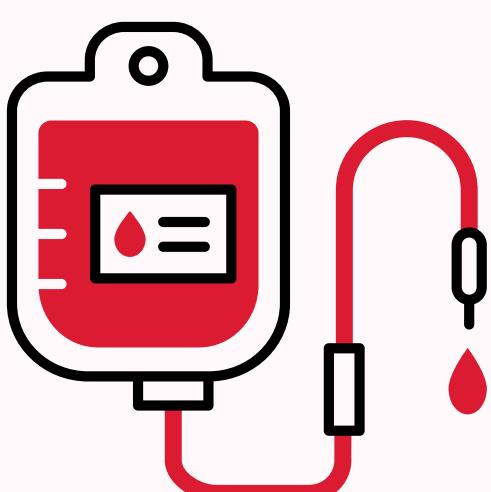
```

1 • SELECT * FROM newschema.blood_transfusion;
2 • SELECT DISTINCT(quantity)
3   FROM newschema.blood_transfusion;
  
```

The Result Grid displays the distinct values from the 'quantity' column: 3.8, 1.3, 1.6, 1.9, 2, 3.2, 2.1, and 2.4.

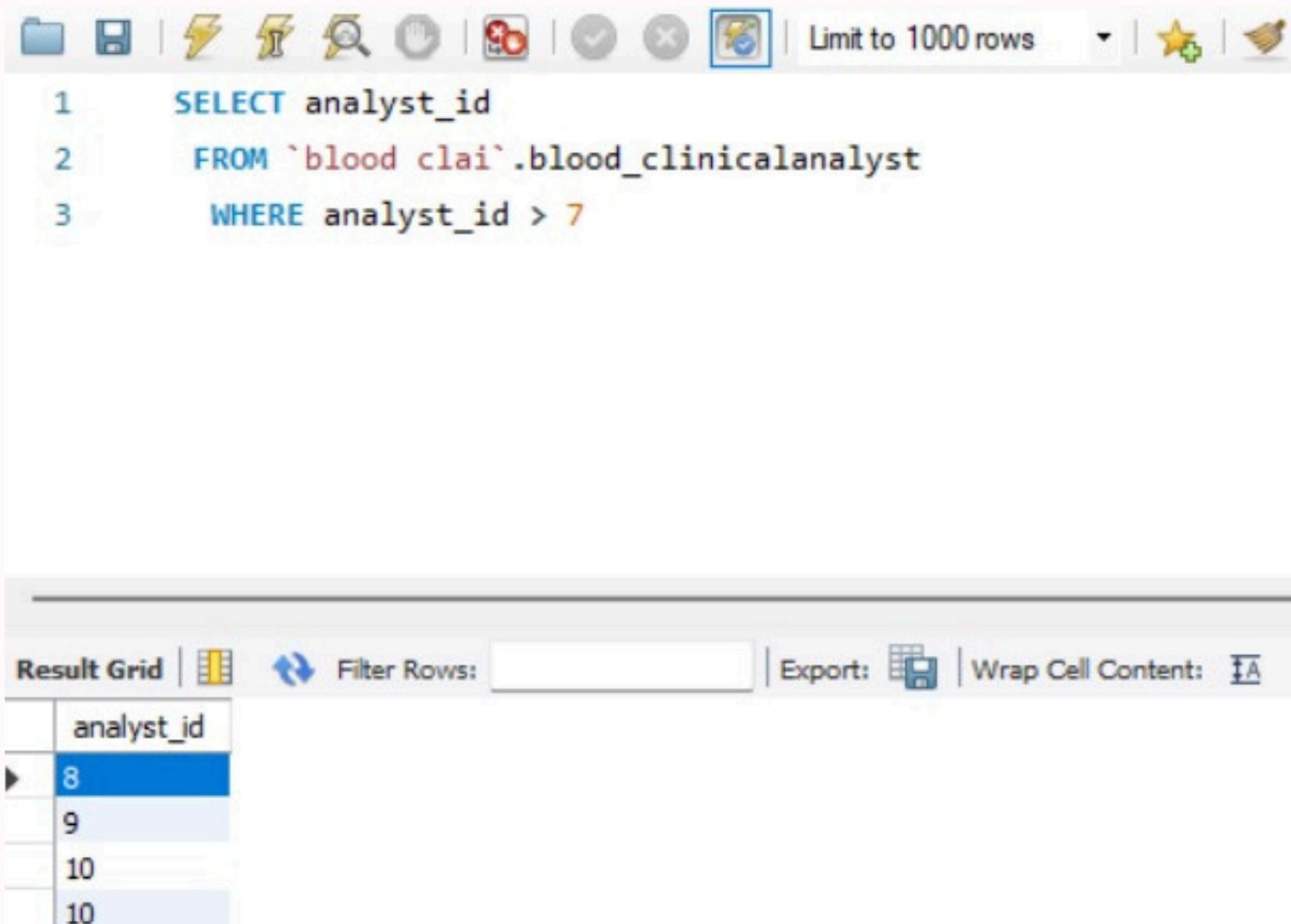
quantity
3.8
1.3
1.6
1.9
2
3.2
2.1
2.4

By Reem



# QUERYS

- Find all blood clinicalanalyst whose analyst\_id is greater than 7



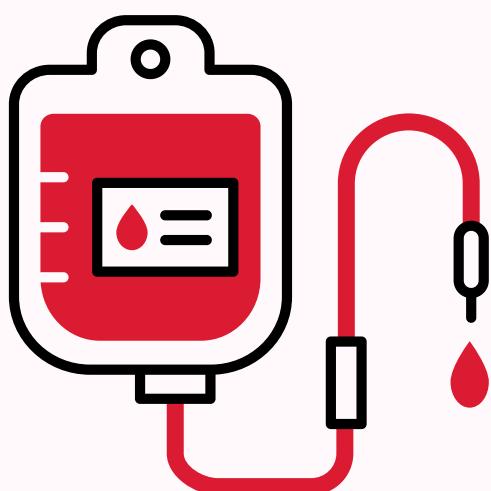
The screenshot shows a MySQL Workbench interface. The query editor contains the following SQL code:

```
1 SELECT analyst_id
2 FROM `blood clai`.blood_clinicalanalyst
3 WHERE analyst_id > 7
```

The result grid displays the following data:

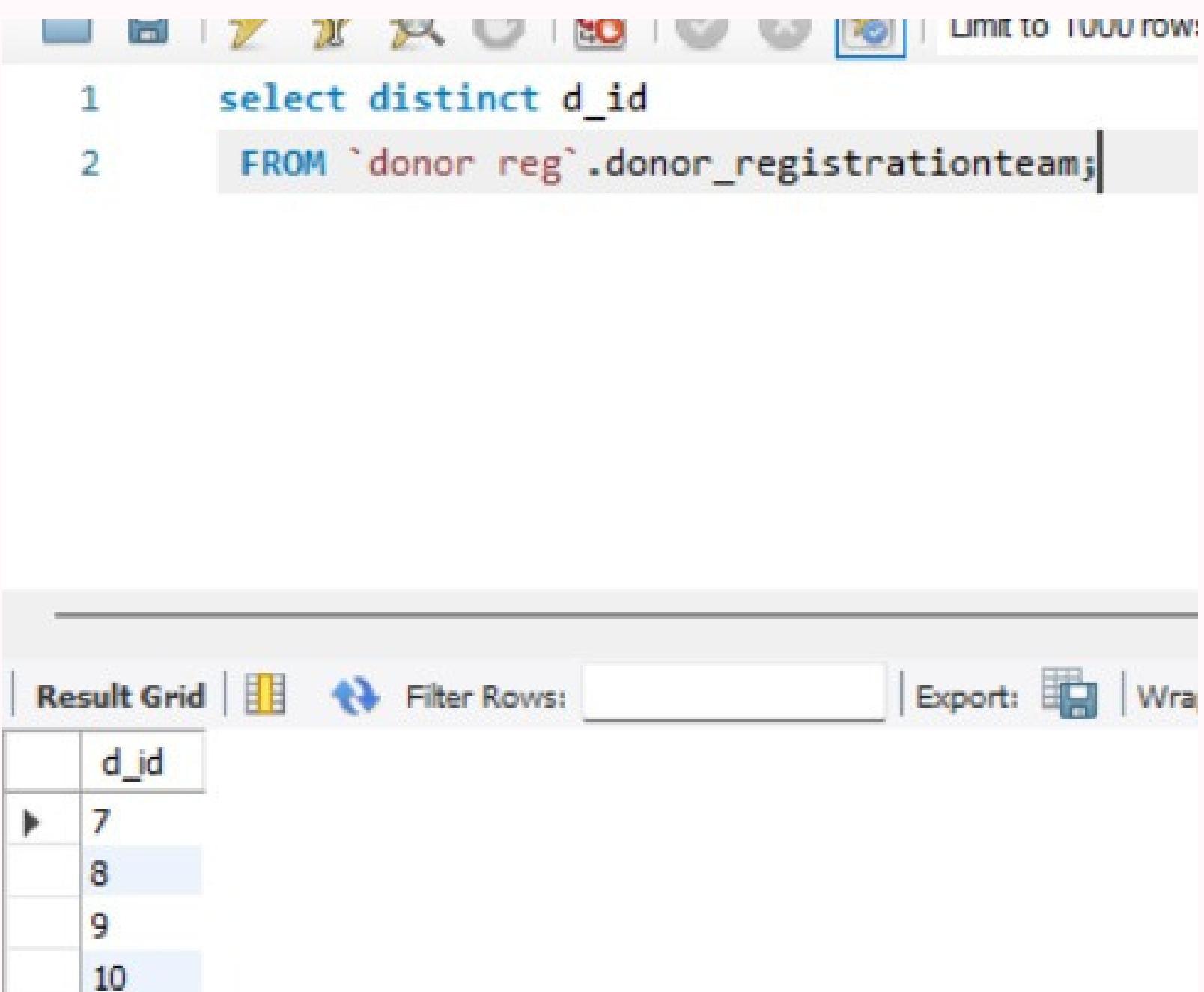
analyst_id
8
9
10
10

By Ghadeer



# QUERYS

- Using DISTINCT to remove the duplicate rows



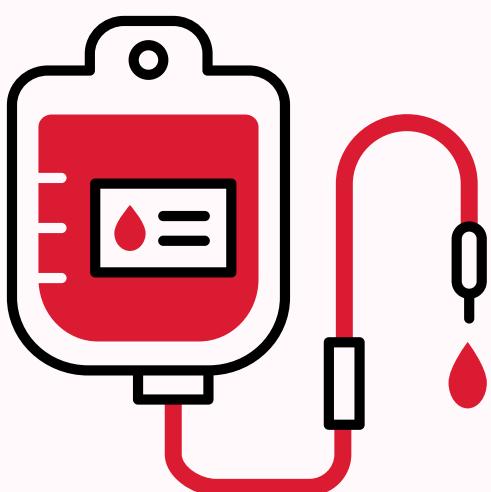
The screenshot shows a MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a query editor window containing the following SQL code:

```
1 select distinct d_id
2 FROM `donor reg`.donor_registrationteam;
```

Below the query editor is a results grid titled "Result Grid". It has one column labeled "d\_id" and four rows with values 7, 8, 9, and 10.

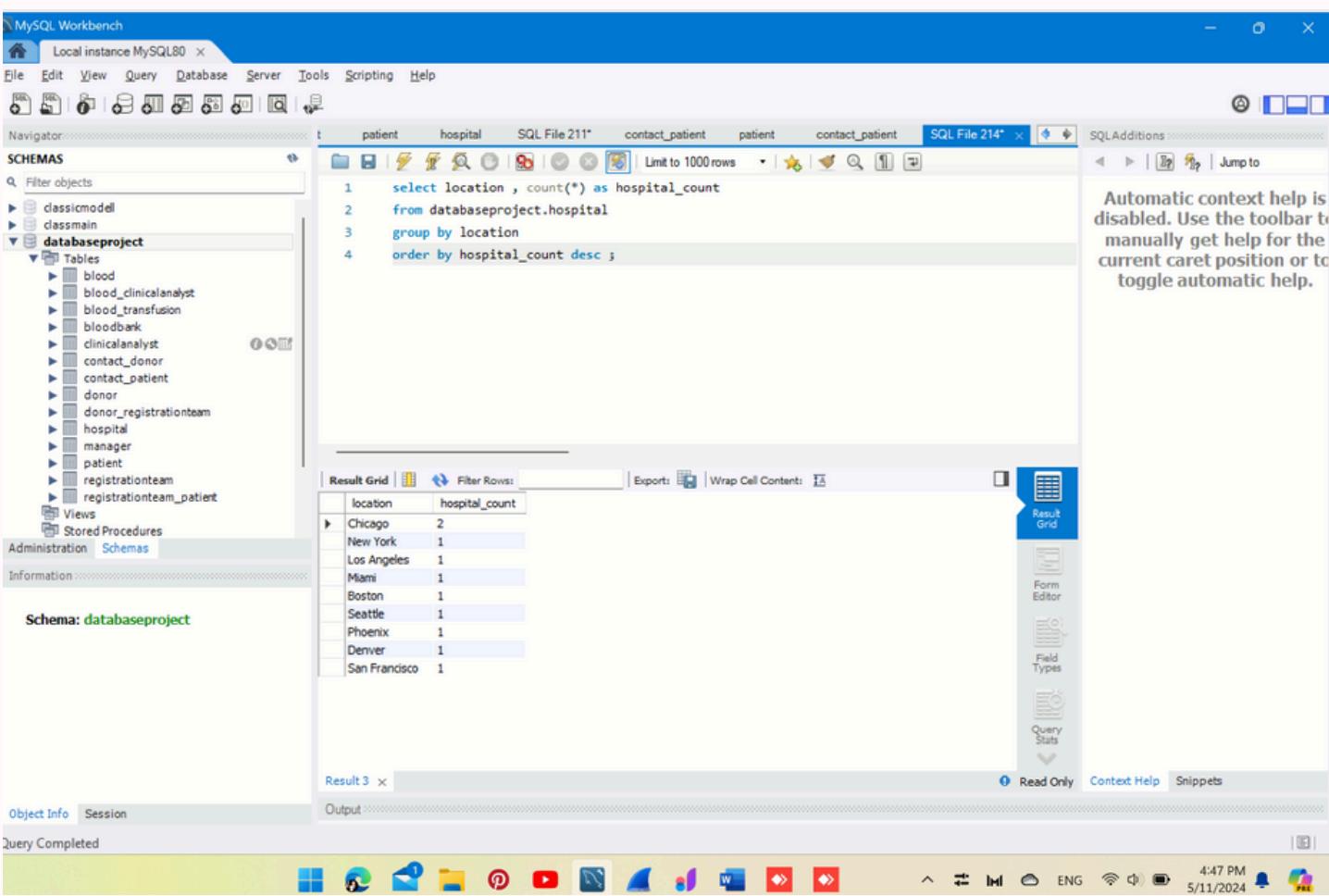
d_id
7
8
9
10

By Ghadeer



# QUERYS

- counts the number of hospitals in each location and orders the results by the count of hospitals in descending order



```

MySQL Workbench - Local Instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator: patient hospital SQL File 211* contact_patient patient contact_patient SQL File 214* 
Schemas: databaseproject
Tables: blood blood_clinicalanalyst blood_transfusion bloodbank clinicalanalyst contact_donor contact_patient donor donor_registrationteam hospital manager patient registrationteam registrationteam_patient Views Stored Procedures Administration Schemas
Information: Schema: databaseproject
Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid
1. select location , count(*) as hospital_count
2. from databaseproject.hospital
3. group by location
4. order by hospital_count desc ;
Result 3 x
Object Info Session Output: Read Only Context Help Snippets
Query Completed
  
```

The screenshot shows the MySQL Workbench interface. In the left sidebar, the 'Schemas' section is expanded to show 'databaseproject'. Under 'Tables', there are many entries like 'blood', 'blood\_clinicalanalyst', etc. The main area contains a SQL editor window with the following query:

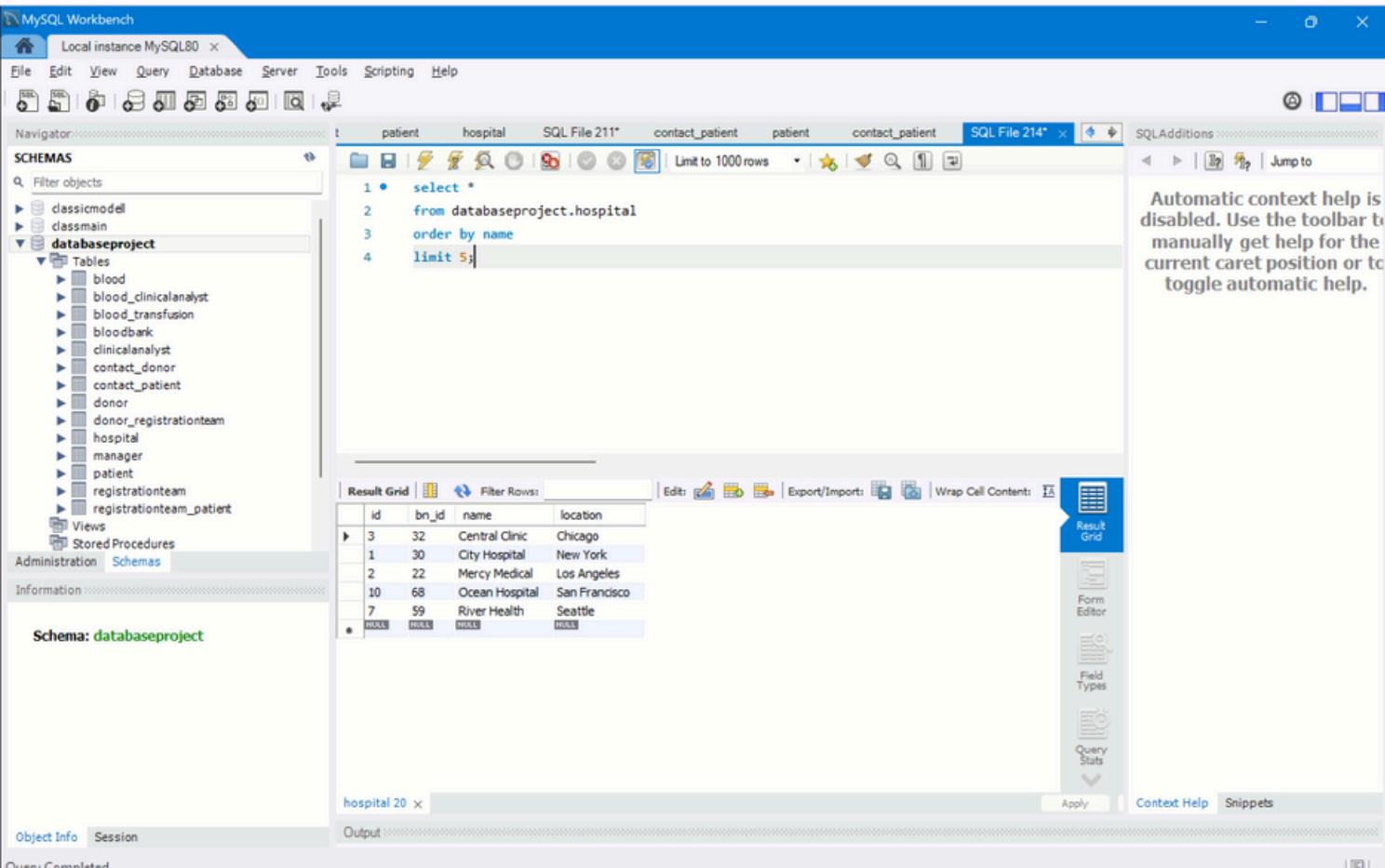
```

1. select location , count(*) as hospital_count
2. from databaseproject.hospital
3. group by location
4. order by hospital_count desc ;
  
```

Below the SQL editor is a 'Result Grid' window displaying the results of the query:

location	hospital_count
Chicago	2
New York	1
Los Angeles	1
Miami	1
Boston	1
Seattle	1
Phoenix	1
Denver	1
San Francisco	1

- all columns from the "Hospital" table, orders the results by the "name" column in ascending order, and limits the output to the first 5 rows



```

MySQL Workbench - Local Instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator: patient hospital SQL File 211* contact_patient patient contact_patient SQL File 214* 
Schemas: databaseproject
Tables: blood blood_clinicalanalyst blood_transfusion bloodbank clinicalanalyst contact_donor contact_patient donor donor_registrationteam hospital manager patient registrationteam registrationteam_patient Views Stored Procedures Administration Schemas
Information: Schema: databaseproject
Result Grid | Filter Rows: Edit: Export/Imports: Wrap Cell Content: Result Grid
1. select *
2. from databaseproject.hospital
3. order by name
4. limit 5;
Result 20 x
Object Info Session Output: Apply Context Help Snippets
Query Completed
  
```

The screenshot shows the MySQL Workbench interface. In the left sidebar, the 'Schemas' section is expanded to show 'databaseproject'. Under 'Tables', there are many entries like 'blood', 'blood\_clinicalanalyst', etc. The main area contains a SQL editor window with the following query:

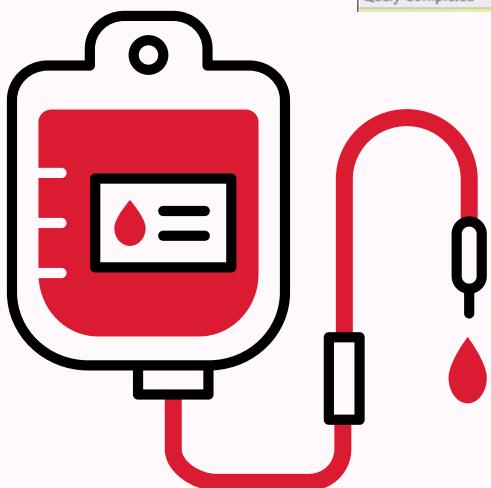
```

1. select *
2. from databaseproject.hospital
3. order by name
4. limit 5;
  
```

Below the SQL editor is a 'Result Grid' window displaying the results of the query:

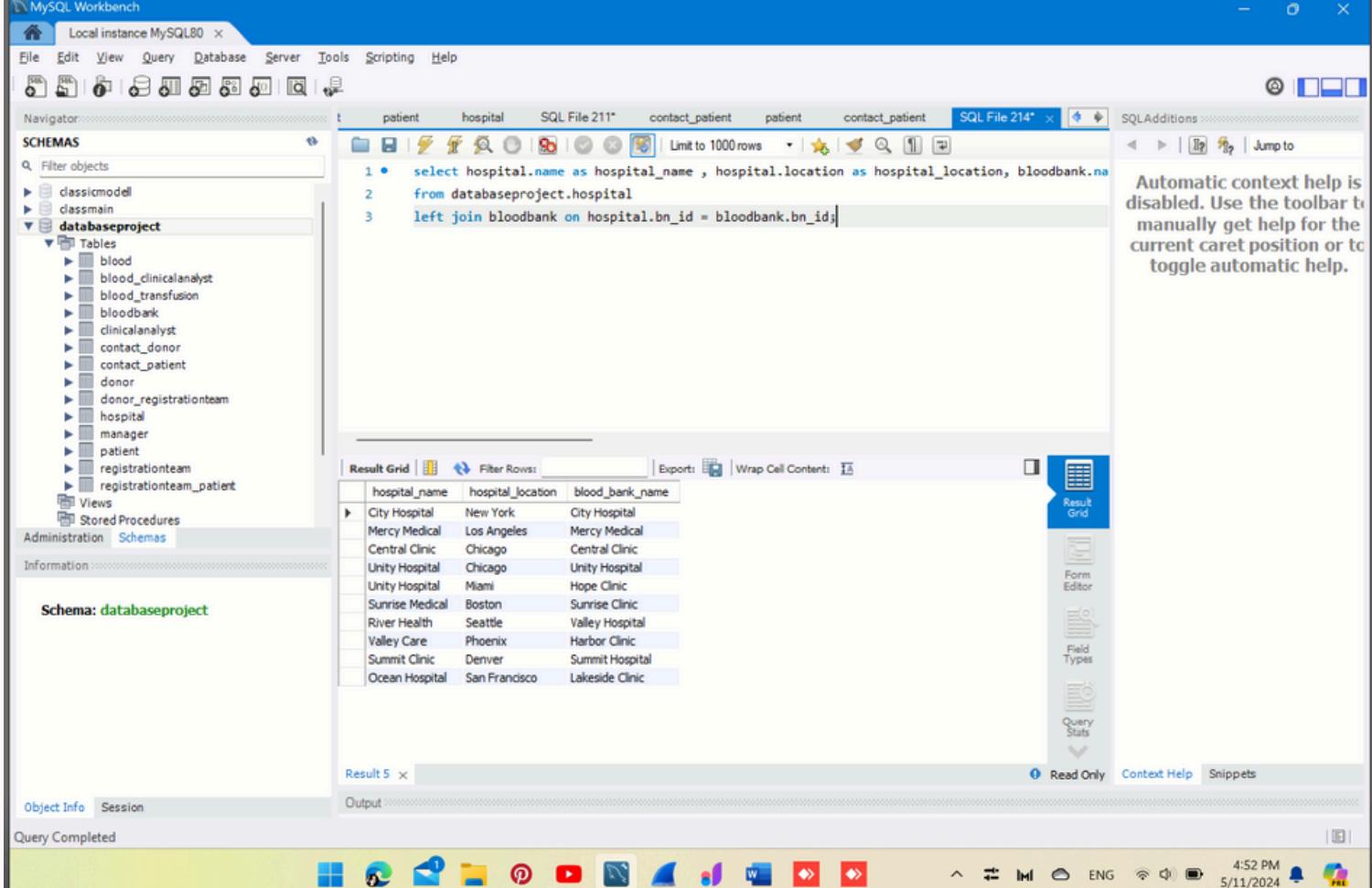
id	bn_id	name	location
3	32	Central Clinic	Chicago
1	30	City Hospital	New York
2	22	Mercy Medical	Los Angeles
10	68	Ocean Hospital	San Francisco
7	59	River Health	Seattle

By shahad



# QUERYS

- this code gets hospital names with associated bank names by matching their locations using LEFT JOIN



The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query is:

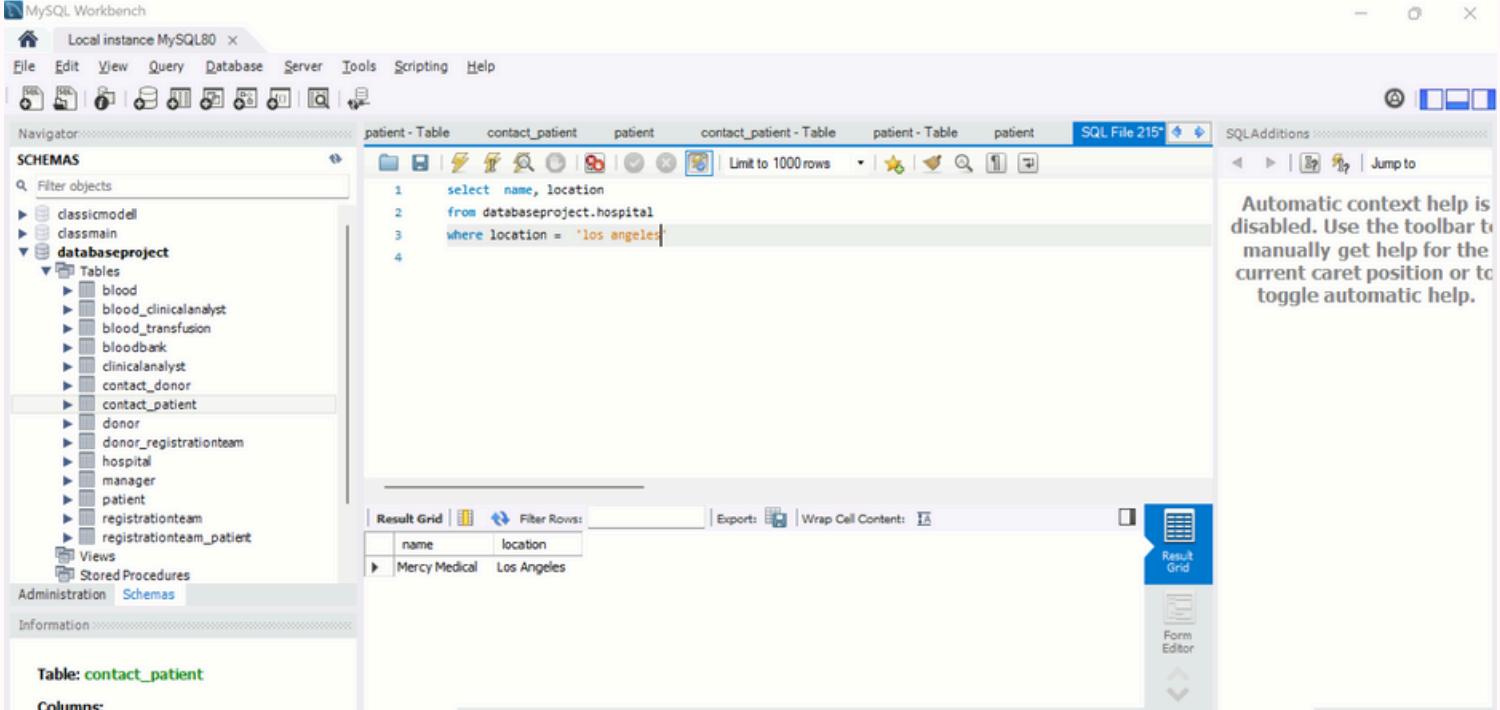
```

1 • select hospital.name as hospital_name , hospital.location as hospital_location, bloodbank.name
2   from databaseproject.hospital
3   left join bloodbank on hospital.bn_id = bloodbank.bn_id
  
```

The result grid displays the following data:

hospital_name	hospital_location	blood_bank_name
City Hospital	New York	City Hospital
Mercy Medical	Los Angeles	Mercy Medical
Central Clinic	Chicago	Central Clinic
Unity Hospital	Chicago	Unity Hospital
Unity Hospital	Miami	Hope Clinic
Sunrise Medical	Boston	Sunrise Clinic
River Health	Seattle	Valley Hospital
Valley Care	Phoenix	Harbor Clinic
Summit Clinic	Denver	Summit Hospital
Ocean Hospital	San Francisco	Lakeside Clinic

- This query fetches hospital names and their locations where the location is 'Los Angeles'



The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query is:

```

1 select name, location
2   from databaseproject.hospital
3  where location = "los angeles"
  
```

The result grid displays the following data:

name	location
Mercy Medical	Los Angeles

By shahad

