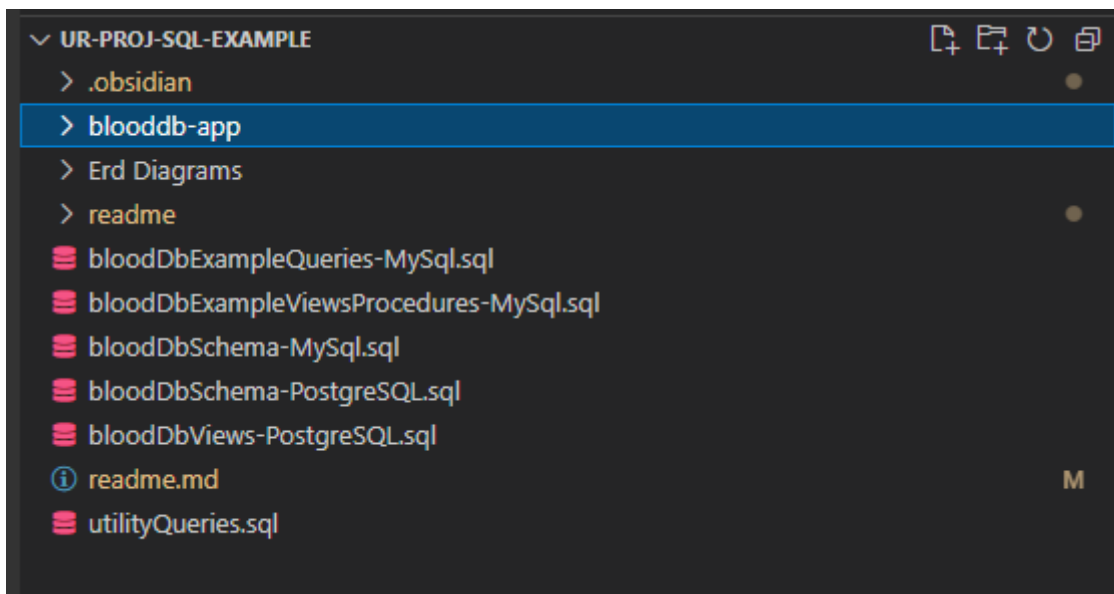


# Blood Transfusion Database Example with simple CRUD App

This codebase consists of SQL schemas and example queries of few tables that provides simplified blood/donor/patient/hospital/transfusion records management system (Mysql version) and simple CRUD PHP app based on Materialized Views and triggers for CS (Computer Science) education purposes.

## Project Files



)

## MySQL version

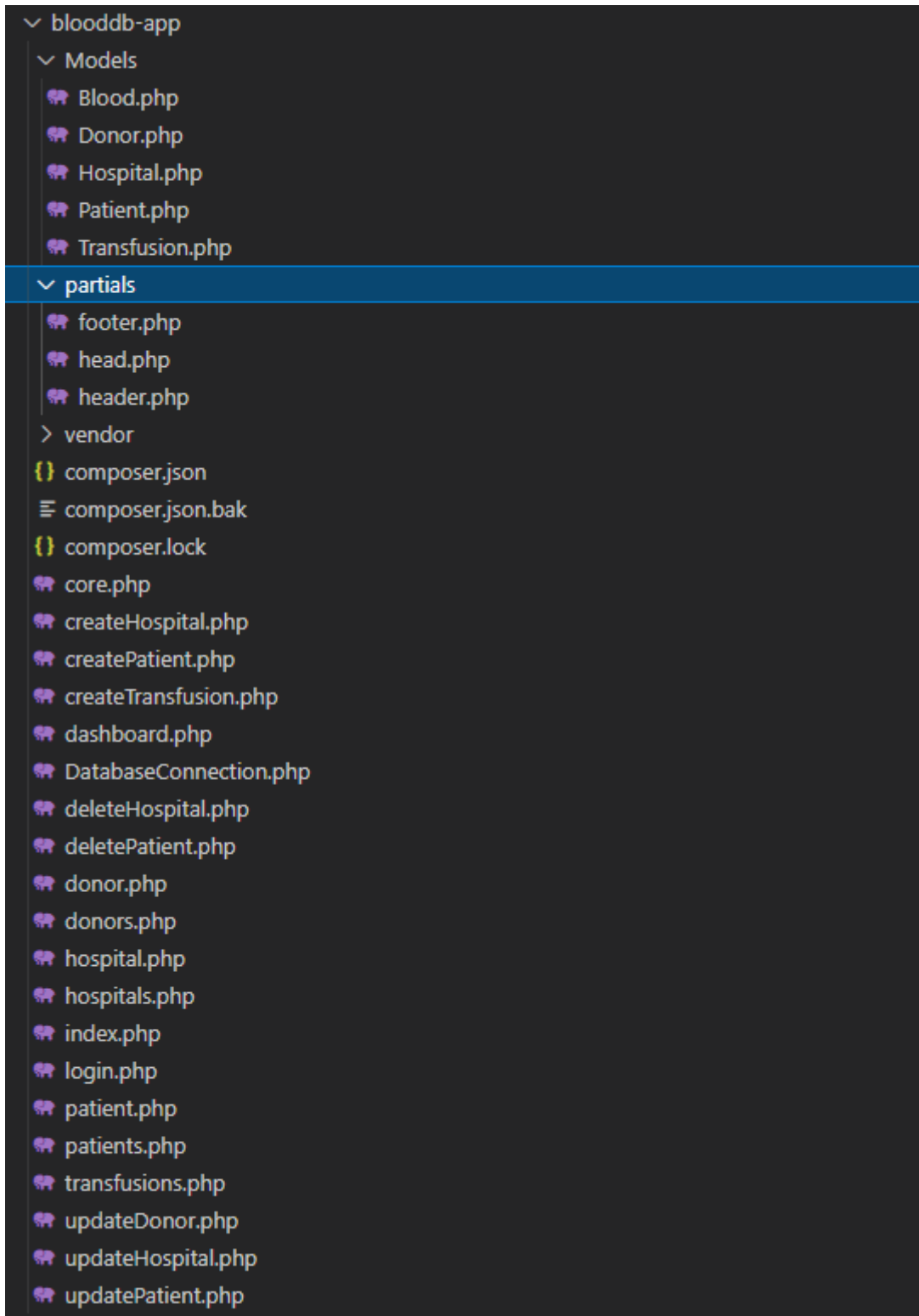
- bloodDbSchema-MySql.sql
- bloodDbExampleQueries-MySql.sql
- bloodDbViewsProcedures-MySql.sql

## Postgres version (modified MySQL version)

Designed for blooddb-app (contains DDL DDM DQL TCL commands):

- bloodDbSchema-Postgres.sql

## Blooddb-app

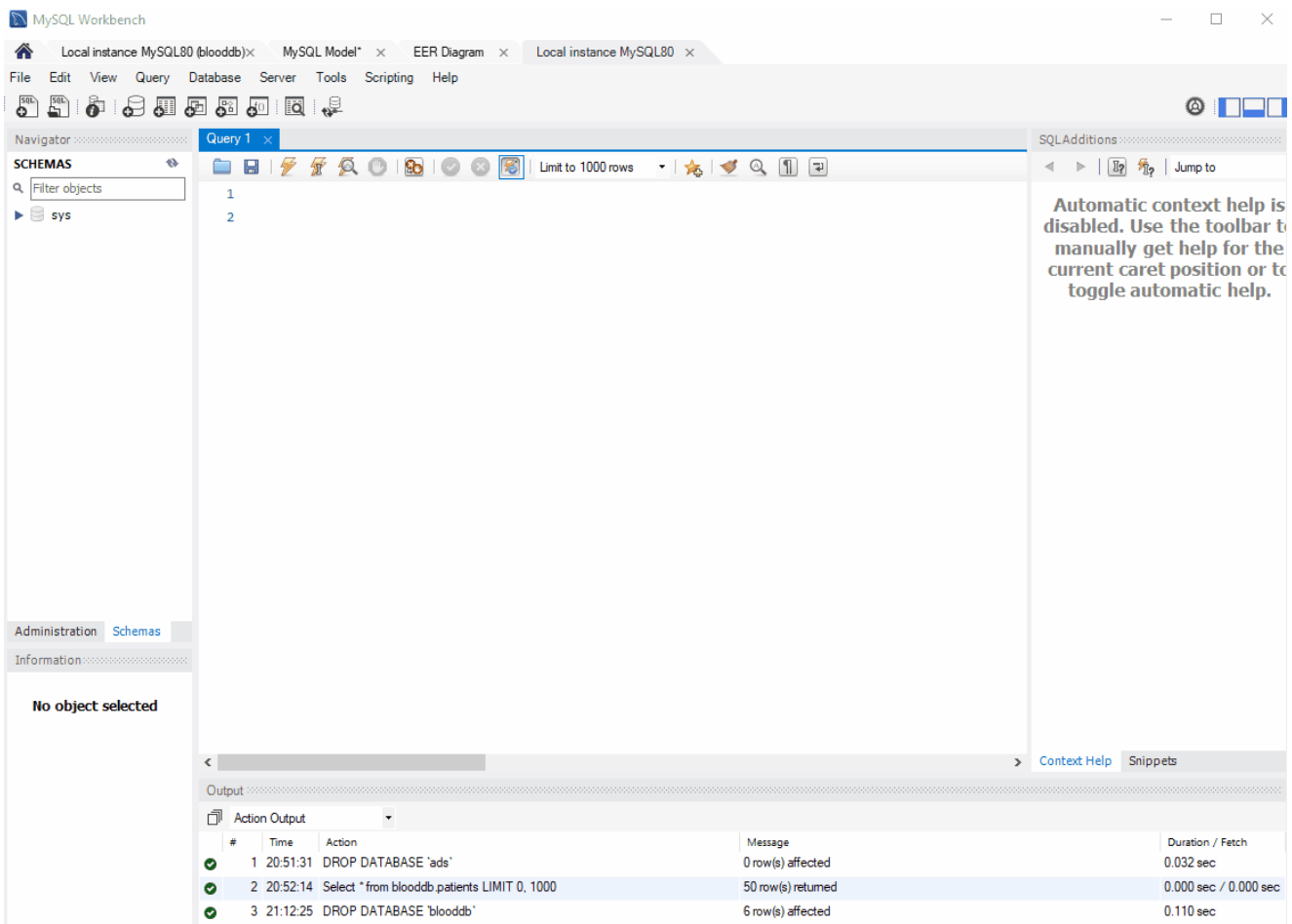


)

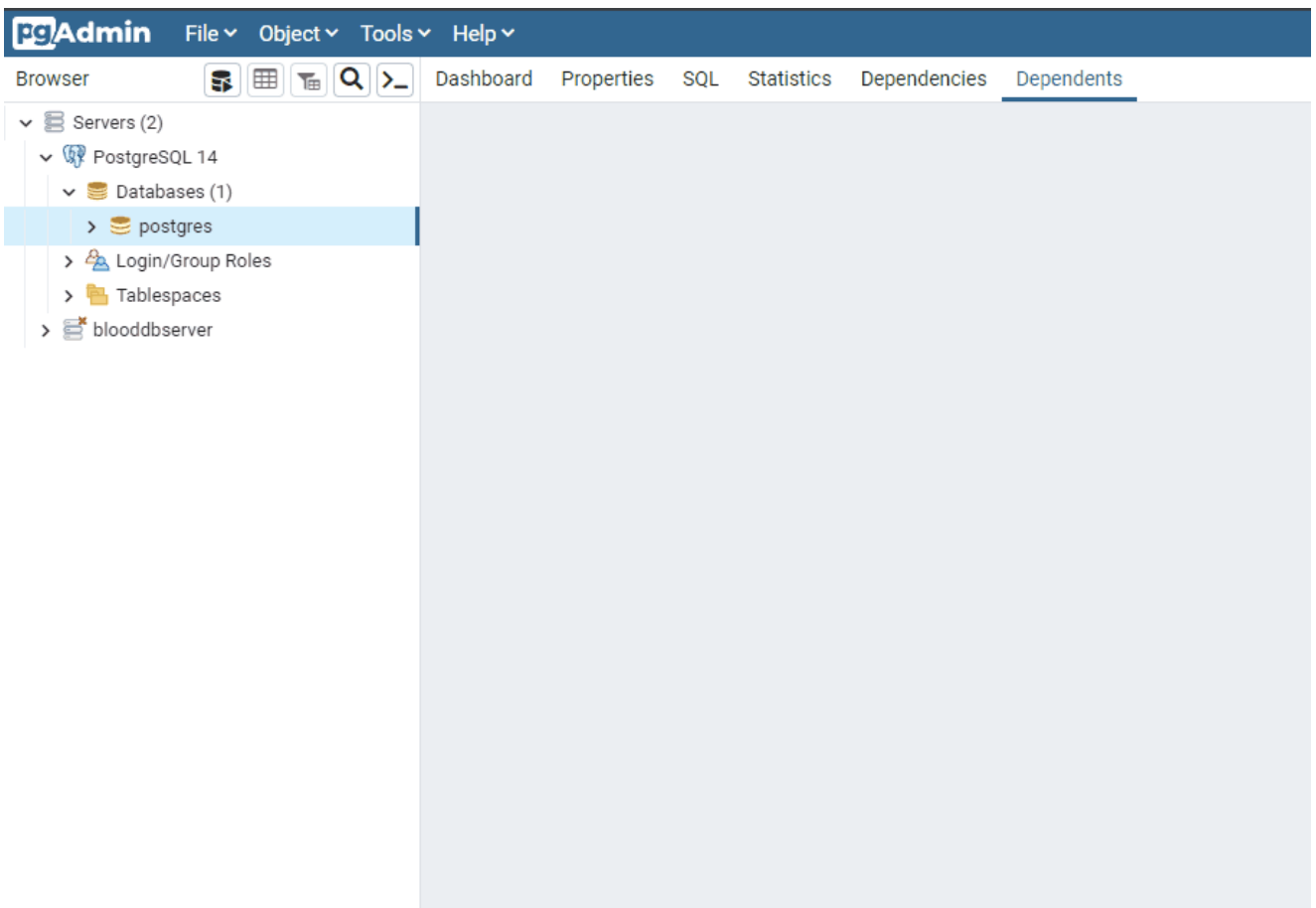
## How to import schema and create query:

---

### MySql Workbench:



## PGAdmin v4:



# How to run blooddb-app

Prerequisites:

- PHP: 7.4 with Composer
- Postgres server: 14

Import Postgres schema and run composer init and php server in blooddb-app directory:

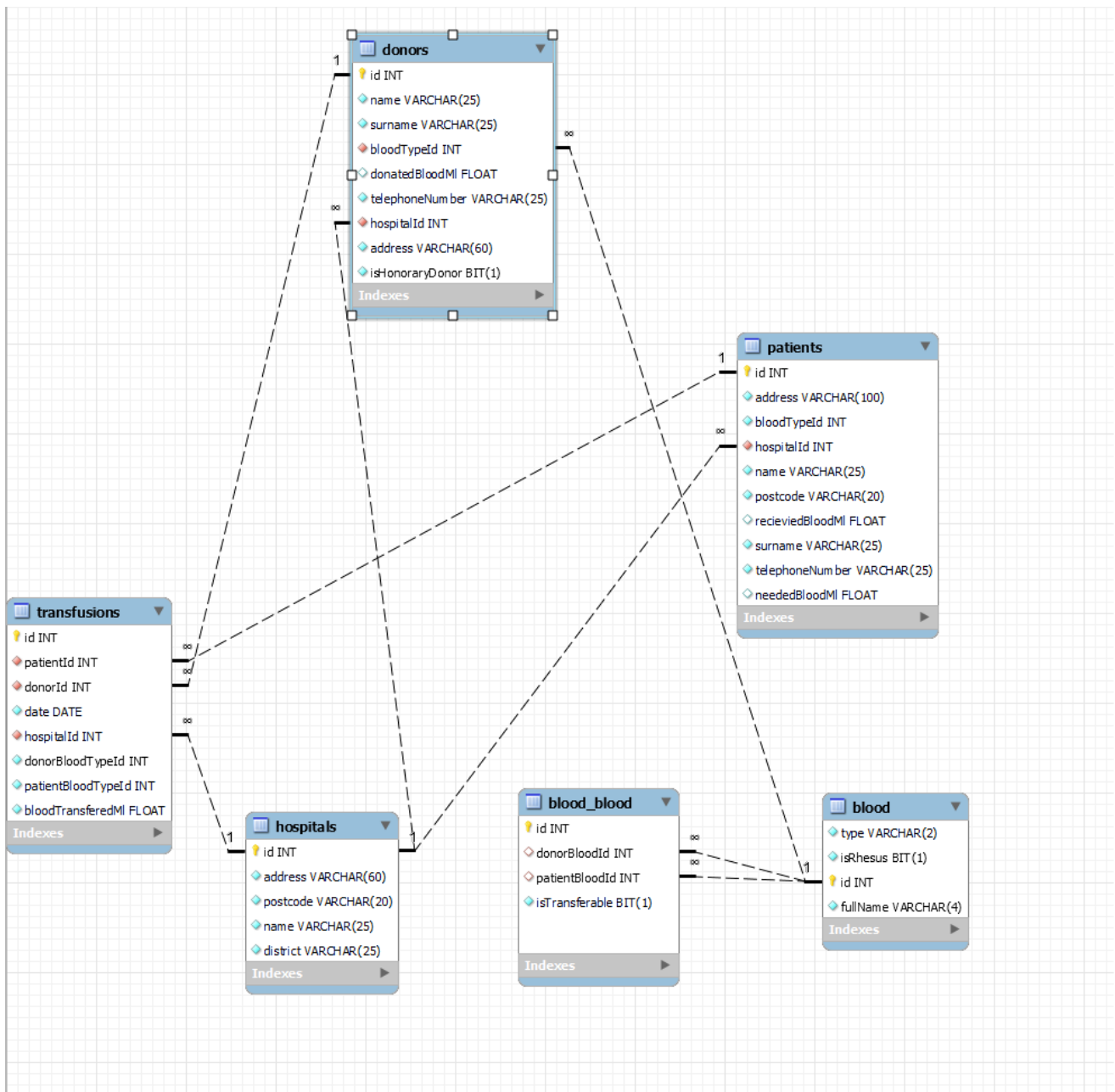
```
/ur-proj-sql-example/blooddb-app (main)  
$ php -S localhost:8020
```

```
/ur-proj-sql-example/blooddb-app (main)  
$ composer install
```

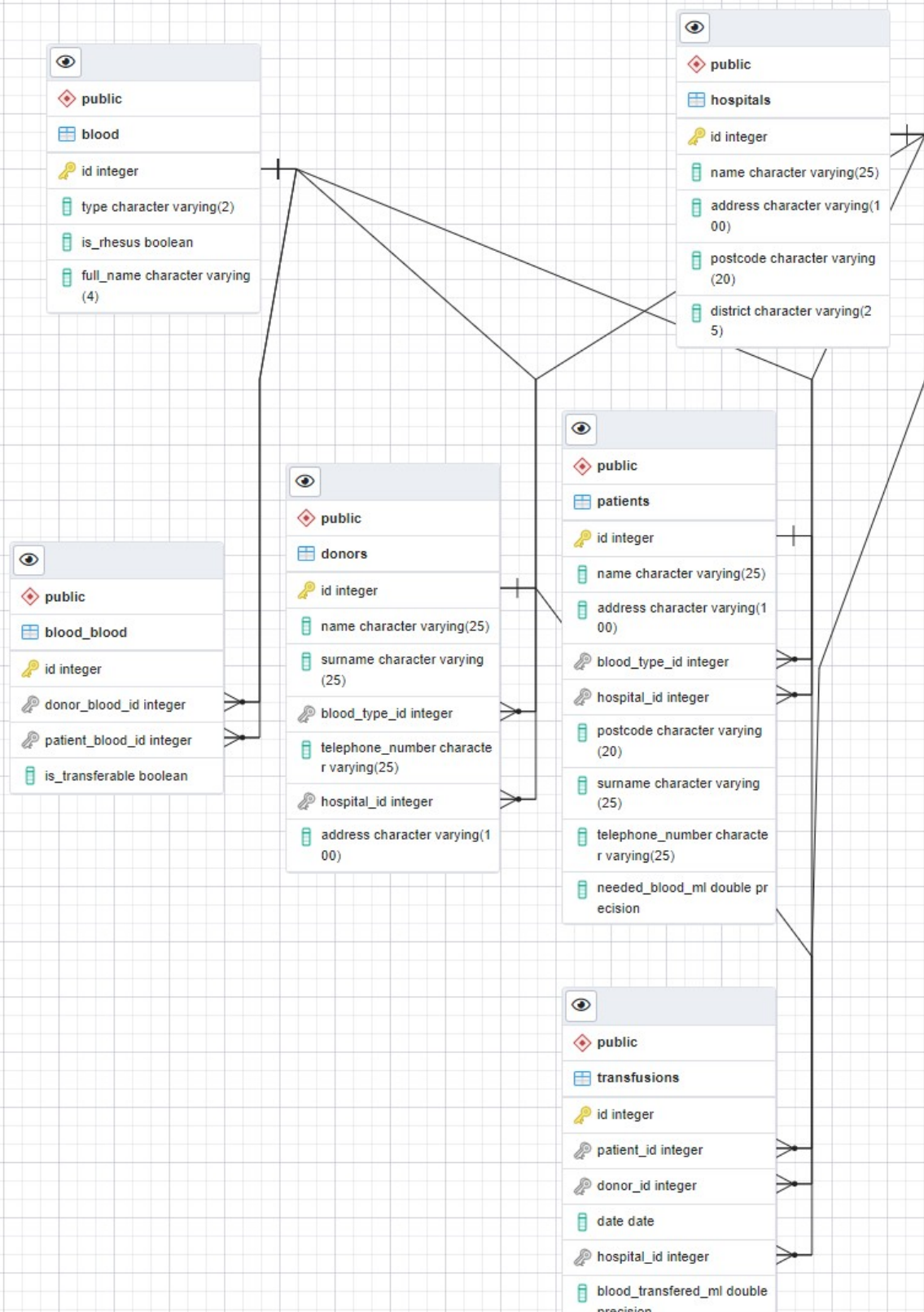
## Schema overview

---

### Mysql Version



## Postgres Version



BloodDb schema consists of:

- 6 tables
- 1 historical table can be generated by "MoveToOldtransfusions" procedure
- 6 one to many relations
- 2 many to many relations

# Tables and relations overview

---

## Blood

```
CREATE TABLE blood (  
  type character varying(2) NOT NULL,  
  isRhesus bit NOT NULL,  
  id integer UNIQUE NOT NULL PRIMARY KEY auto_increment,  
  fullName character varying(4) NOT NULL  
);
```

	type	isRhesus	id	fullName
▶	AB	1	1	AB+
	AB	0	2	AB-
	A	1	3	A+
	A	0	4	A-
	B	1	5	B+
	B	0	6	B-
	O	1	7	O+
	O	0	8	O-
*	NULL	NULL	NULL	NULL

Table represents all possible combinations of antigens and D antibodies of human blood

ABO is the major blood group system and is genetically determined. ABO type depends on A or B antigens on the red cells. If blood type is O, cells don't have either type of antigen.

D is the most important antigen of the Rh (Rhesus) system and it's also known as RhD.

Blood Ids from this table are assigned to patients and donors.(one to many)

Fields:

- id - autoincremented PK
- type - blood ABO group
- isRhesus - true/false value about RhD antigen
- fullName - full name of blood type, combines values from "type" and "isRhesus" fields

## Blood\_Blood

---

```
CREATE TABLE blood_blood (
    id INT UNIQUE NOT NULL PRIMARY KEY auto_increment,
    donorBloodId integer,
    patientBloodId integer,
    isTransferable bit NOT NULL
);
```

	id	donorBloodId	patientBloodId	isTransferable
▶	1	2	1	1
	2	1	1	1
	3	3	1	1
	4	4	1	1
	5	5	1	1
	6	6	1	1
	7	7	1	1
	8	8	1	1
	9	1	2	0
	10	2	2	1

Before transfusion, testing is done between the donated blood and patient to check for compatibility.

Table contains blood group compatibility table :

Recipient <sup>[1]</sup>	Donor <sup>[1]</sup>							
	O-	O+	A-	A+	B-	B+	AB-	AB+
O-	✓	✗	✗	✗	✗	✗	✗	✗
O+	✓	✓	✗	✗	✗	✗	✗	✗
A-	✓	✗	✓	✗	✗	✗	✗	✗
A+	✓	✓	✓	✓	✗	✗	✗	✗
B-	✓	✗	✗	✗	✓	✗	✗	✗
B+	✓	✓	✗	✗	✓	✓	✗	✗
AB-	✓	✗	✓	✗	✓	✗	✓	✗
AB+	✓	✓	✓	✓	✓	✓	✓	✓

Table in schema is implemented as many to many relation between same table : "blood"

Technically "blood\_blood" table looks like transfer table

Fields

- id - autoincremented PK
- donorBloodID - blood id from blood table (many to many Foreign Key)
- patientBloodID - blood id from blood table (many to many Foreign Key)
- isTransferable - stores true/false value with provides info about compatibility between donor and recipient blood



# Hospitals

---

```
CREATE TABLE hospitals (  
    id INT UNIQUE NOT NULL PRIMARY KEY auto_increment,  
    address character varying(60) NOT NULL,  
    postcode character varying(20) NOT NULL,  
    name character varying(25) NOT NULL,  
    district character varying(25) NOT NULL  
);
```

	id	address	postcode	name	district
▶	1	4960 Macpherson Cirde	33-115	St. Mary Hospital	West
	2	2 Fairview Drive	33-115	Main West Hospital	West
	3	93 Ryan Plaza	33-115	Giga Hospital	East
	4	73567 Grim Junction	503-1305	St. John Hospital	East
*	NULL	NULL	NULL	NULL	NULL

This simple table stores all registered hospitals. All patients and Donors are assigned by one to many relation to the one of hospital. Transfusion table is also connected by the same relation.

## Fields

- id - autoincremented PK
- address - Hospital address
- postcode- Hospital postcode
- name - full name of hospital
- district -city district of hospital

# Patients

---

```
CREATE TABLE patients (  
    id INT UNIQUE NOT NULL PRIMARY KEY auto_increment,  
    address character varying(100) NOT NULL,  
    bloodTypeId INT NOT NULL,  
    hospitalId INT NOT NULL,  
    name character varying(25) NOT NULL,  
    postcode character varying(20) NOT NULL,  
    recievedBloodMl float,  
    surname character varying(25) NOT NULL,  
    telephoneNumber character varying(25) NOT NULL,  
    neededBloodMl float  
);
```

	id	address	bloodTypeId	hospitalId	name	postcode	recievedBloodMl	surname	telephoneNumber	neededBloodMl
▶	1	959 Grasskamp Way	4	4	Maren	76152	3979	Benford	406-732-1702	NULL
	2	15 Hooker Trail	3	3	Beverley	76152	1076	Dudgeon	297-933-2937	NULL
	3	65690 Orin Plaza	3	1	Veronika	76152	612	Bosanko	845-547-1203	NULL
	4	9 Green Ridge Way	4	1	Antonino	6344	580	Murum	778-518-1019	NULL
	5	38789 Linden Junction	8	1	Lynelle	6344	942	Fitchet	503-444-4863	342
	6	9805 Homewood Hill	3	1	Katha	06-413	85	Tierney	903-222-7909	1450
	7	0789 Annamark Plaza	1	4	Cathlene	4700-837	1152	Mathon	530-252-2150	NULL
	8	268 Norway Maple Circle	5	1	Cristobal	4700-838	3006	Yoseloff	917-701-0053	NULL
	9	718 Shopko Terrace	5	3	Julienne	4700-839	1014	Pascoe	128-518-0762	NULL
	10	181 Anhalt Court	8	1	Mina	4700-840	218	Snead	865-412-3372	NULL

Stores data about patients. Table is related with transfusions, hospitals, blood table by one to many relation

### Fields

- id - autoincremented PK
- address - Patient address
- postcode- Patient postcode
- surname - Patient surname
- name - Patient name
- bloodTypeId - assigned blood type id from blood table
- **MySQL Schema only:** recievedBloodMl - int value of recieved blood in Ml (sum of blood from all transfusions)
- telephone number - telephone number
- neededBloodMl - float value of neededBlood in Ml
- hospitalId - id of assigned hospital

## Donors

Stores data about donors. Table is related with transfusions,hospitals, blood table by one to many relation

```
CREATE TABLE donors (
  id int NOT NULL UNIQUE PRIMARY KEY auto_increment,
  name character varying(25) NOT NULL,
  surname character varying(25) NOT NULL,
  bloodTypeId integer NOT NULL,
  donatedBloodMl float,
  telephoneNumber character varying(25) NOT NULL,
  hospitalId integer NOT NULL,
  address character varying(60) NOT NULL,
  isHonoraryDonor bit NOT NULL
);
```

	id	name	surname	bloodTypeId	donatedBloodMl	telephoneNumber	hospitalId	address	isHonoraryDonor
▶	1	Margalo	Vaskin	8	58	+7 (830) 277-4012	3	658 Forest Center	0
	2	Kordula	McAlindon	7	2353	+1 (941) 846-8640	3	0531 Chinook Center	0
	3	Goldina	Tesimon	7	3413	+54 (103) 401-2327	1	9772 Menomonie Plaza	1
	4	Lina Goldina	athys	7	946	+33 (200) 228-3321	2	86 Dawn Pass	0
	5	Eric	Ickovits	7	95	+1 (414) 422-0805	1	6746 Loftsgordon Court	0
	6	Urbanus	Gidney	1	2071	+86 (148) 208-4087	1	80 Marquette Junction	0
	7	Darla	Gors	6	2028	+86 (841) 691-2186	3	45 Lotheville Plaza	0
	8	Lauritz	Dinnage	3	1344	+60 (445) 479-4532	1	19209 Anthes Point	0
	9	Harriot	Sitford	4	1596	+30 (684) 716-3053	3	82 Farmco Pass	0
	10	Adrienne	Becket	2	3400	+86 (818) 745-7308	3	7 Caliangt Terrace	1

## Fields

- id - autoincremented PK
- address - donor address
- postcode- donor postcode
- surname - donor surname
- name - donor name
- bloodTypeId - assigned blood type id from blood table
- **MySQL Schema only:** donatedBloodMl - int value of recieved blood in Ml (sum of blood from all transfusions)
- telephone number - telephone number
- hospitalId - id of assigned hospital
- **MySQL Schema only:** isHonoraryDonor - true false value whitch determines if donor is honorary donor (donated blood is equal or higher than 3000 ml )

## Transfusions

Holds records about transfusions between patients and donors in specific hospital (one to many ).

```
CREATE TABLE transfusions (
  id integer UNIQUE NOT NULL PRIMARY KEY auto_increment,
  patientId integer NOT NULL,
  donorId integer NOT NULL,
  date date NOT NULL,
  hospitalId integer NOT NULL,
  bloodTransferredMl float NOT NULL
);
```

	id	patientId	donorId	date	hospitalId	bloodTransferredML
▶	2	30	26	2021-08-02	3	424
	4	2	19	2021-10-13	4	1076
	6	35	1	2021-08-28	1	58
	8	4	24	2021-09-22	2	580
	9	27	15	2021-07-28	3	1066
	13	50	43	2021-09-21	2	751
	18	35	41	2021-07-18	2	3794
	22	14	18	2021-12-21	3	2022
	23	50	40	2021-07-26	1	3680
	25	16	46	2021-11-04	1	3137

- id - autoincremented PK
- patientID - id patient who recieved blood
- donorID - id donor who donated blood
- hospitalId - id of hospital where transfusion had been done
- date - date of transfusion in ' YYYY-MM-DD ' format
- bloodTransferredML - float value of transfered blood during transfusion in ML

By using procedure "MoveToOldtransfusions" as CRON job (executes commands at specific dates and times) transfusion table can behave like a temporary table with the newest records (for ex. recent month). Older transfusions would be stored in old\_transfusions table.

```
CREATE PROCEDURE `MoveToOldtransfusions`()
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        RESIGNAL;
    END;
    CREATE TABLE IF NOT EXISTS oldTransfusions (
        id integer UNIQUE NOT NULL PRIMARY KEY auto_increment,
        patientId integer NOT NULL,
        donorId integer NOT NULL,
        date date NOT NULL,
        hospitalId integer NOT NULL,
        bloodTransferredML float NOT NULL
    );
    START TRANSACTION;
    insert into oldTransfusions
    Select transfusions.* from transfusions
    where transfusions.date < DATE("2022-01-15" - INTERVAL 1 MONTH);
    delete from transfusions where transfusions.date < DATE("2022-01-15" - INTERVAL 1 MONTH);
    COMMIT;
END
```

## Relations

All mentioned relations with foreign keys are defined below

```

ALTER TABLE donors
ADD FOREIGN KEY (hospital_id) REFERENCES hospitals (id) ON DELETE CASCADE;
ALTER TABLE donors
ADD FOREIGN KEY (blood_type_id) REFERENCES blood (id) ON DELETE CASCADE;
ALTER TABLE blood_blood
ADD FOREIGN KEY (donor_blood_id) REFERENCES blood(id) ON DELETE CASCADE;
ALTER TABLE blood_blood
ADD FOREIGN KEY (patient_blood_id) REFERENCES blood(id) ON DELETE CASCADE;
ALTER TABLE patients
ADD FOREIGN KEY (hospital_id) REFERENCES hospitals (id) ON DELETE CASCADE;
ALTER TABLE patients
ADD FOREIGN KEY (blood_type_id) REFERENCES blood (id) ON DELETE CASCADE;
ALTER TABLE transfusions
ADD FOREIGN KEY (patient_id) REFERENCES patients (id) ON DELETE CASCADE;
ALTER TABLE transfusions
ADD FOREIGN KEY (donor_id) REFERENCES donors (id) ON DELETE CASCADE;
ALTER TABLE transfusions
ADD FOREIGN KEY (hospital_id) REFERENCES hospitals (id) ON DELETE CASCADE;

```

## Mock data generation steps

Dataset was generated using a [Mockaroo](#) online random data generator

Field Name	Type	Options
id	Row Number	blank: 0 % $\Sigma$ X
first_name	First Name	blank: 0 % $\Sigma$ X
last_name	Last Name	blank: 0 % $\Sigma$ X
email	Email Address	blank: 0 % $\Sigma$ X
gender	Gender	blank: 0 % $\Sigma$ X
ip_address	IP Address v4	blank: 0 % $\Sigma$ X

[ADD ANOTHER FIELD](#)

# Rows: 1000    Format: CSV    Line Ending: Unix (LF)    Include: ☒ header ☐ BOM

It can export generated data to CSV, SQL format.

Some tables like "transfusions" needed to be verified for example in terms of blood type compatibility. Also transferred blood and

"honorary donor" value must be related with data from other tables. It was accomplished by using some alter or delete and some "utility" queries on live database.

Utility Queries (Delete, Update) used to generate consistent dataset :

```
delete transfusions from transfusions
inner join patients on patients.id = transfusions.patientId
inner join donors on donors.id = transfusions.donorId
inner join blood_blood on donors.bloodTypeId = blood_blood.donorBloodId
and patients.bloodTypeId = blood_blood.patientBloodId
where blood_blood.isTransferable = false

select patients.id ,count(transfusions.bloodTransferredML)  from patients
inner join transfusions on patients.id = transfusions.patientId
group by patients.id

create view "patientsTransferredBloodSummary" as
Select patients.id ,transfusions.bloodTransferredML from patients
inner join transfusions on patients.id = transfusions.patientId
group by patients.id

create view "donorsTransferredBloodSummary" as
Select donors.id ,transfusions.bloodTransferredML from donors
inner join transfusions on donors.id = transfusions.donorId
group by donors.id

UPDATE patients
INNER JOIN patientsTransferredBloodSummary ON patients.id = patientsTransferredBloodSummary.id
SET patients.recievedBloodML = patientsTransferredBloodSummary.bloodTransferredML

UPDATE donors
INNER JOIN donorsTransferredBloodSummary ON donors.id = donorsTransferredBloodSummary.id
SET donors.donatedBloodML = donorsTransferredBloodSummary.bloodTransferredML

UPDATE donors
set donors.isHonoraryDonor = false
where donors.donatedBloodML < 3000

UPDATE donors
set donors.isHonoraryDonor = true
where donors.donatedBloodML >= 3000
```

## Example Queries/Views

---

### 1. Select all patients with blood type AB

---

	sum of patients and donors
▶	9

```
Select patients.* from patients where patients.bloodTypeId IN (Select id from blood where blood.type='AB');
```

### 2.Select all patients and donors with blood group A- and O+ (union example)

---



1 • `SELECT * FROM blooddb1.patientsanddonorswithanegandopos;`

	id	name	surname	bloodTypeId
▶	9	Harriot	Sitford	4
	14	Alix	Davidman	4
	24	Clemente	Fitzsimon	4
	44	Joane	Harvey	4
	49	Red	Kittiman	4
	2	Kordula	McAlindon	7
	3	Goldina	Tesimon	7
	4	Lindon	Mathys	7
	5	Eric	Ickovits	7
	18	Ariadne	Vynarde	7
	20	Tova	Carrett	7
	25	Phineas	Chastenev	7
	26	Chrissie	Willimot	7
	46	Bernhard	Adolthine	7
	1	Maren	Benford	4
	4	Antonino	Murum	4
	13	Grantley	Petrosian	4
	14	Traver	MacMichael	7
	16	Gigi	Kaes	7
	41	Othella	Munnery	4
	44	Karola	Zini	7
	45	Dreddy	Curley	4
	47	Dorie	Craister	7

```
SELECT donors.id ,donors.name, donors.surname, donors.bloodTypeId FROM donors
where donors.bloodTypeId IN (Select id from blood where blood.fullName='A-' or blood.fullName='O+' )
UNION
Select patients.id ,patients.name , patients.surname, patients.bloodTypeId from patients
where patients.bloodTypeId IN (Select id from blood where blood.fullName='A-' or blood.fullName='O+' );
```

### 3.Select all patients that begins with 'a'

"/readme/patientsfemale1.jpg" is not created yet. Click to create.

```
Select patients.* from patients where patients.name like '%a';
```

### 4. Select all patients from hospital of id 2 that their blood was transfused less than 1500 ml

1 • `SELECT * FROM blooddb1.patientslessthan1500recievedbloodfromhospital2;`

	id	address	bloodTypeId	hospitalId	name	postcode	recievedBloodMl	surname	telephoneNumber	neededBloodMl
▶	22	89277 Transport Pass	3	2	Keane	562 64	1482	Le Count	304-305-3439	324
	27	55 Buell Lane	3	2	Billi	567 64	1066	Butting	915-661-3200	NULL
	28	3 8th Park	8	2	Mattie	568 64	879	Snepp	576-790-4180	324
	41	57757 Spenser Trail	4	2	Othella	9800	1249	Munnery	347-303-9047	NULL
	44	87215 Garrison Crossing	7	2	Karola	20210	785	Zini	543-727-2776	NULL
	49	0 Shoshone Crossing	5	2	Ardeen	9505	1488	Pengelly	953-993-3870	333

```
Select patients.* from patients
inner join hospitals on patients.hospitalId = hospitals.id
where hospitals.id = 2 and patients.recievedBloodMl <= 1500
```

## 5. Select transfusions and addresses of hospitals where transfusions had been done between date X to date X

1 • `SELECT * FROM blooddb1.transfusionsbetween20210808and20210809;`




	id	patientId	donorId	date	hospitalId	bloodTransferredMl	address
▶	884	9	43	2021-08-08	2	2209	2 Fairview Drive
	36	6	43	2021-08-08	3	85	93 Ryan Plaza
	788	49	1	2021-08-08	4	659	73567 Grim Junction

```
Select transfusions.*,hospitals.address from hospitals
inner join transfusions on transfusions.HospitalId = hospitals.id
where transfusions.date between "2021-08-08" and "2021-08-09"
```

## 6. Select all transfusions of all hospitals excluding hospital of id 3



1 • SELECT \* FROM blooddb1.transfusionsfromhospital3;

Result Grid			 Filter Rows:	<input type="text"/>	Export: 	Wrap Cell C
	id	patientId	donorId	date	hospitalId	bloodTransferredMl
▶	6	35	1	2021-08-28	1	58
	23	50	40	2021-07-26	1	3680
	25	16	46	2021-11-04	1	3137
	40	32	24	2021-07-18	1	1460
	44	6	41	2021-08-11	1	1140
	59	40	15	2021-12-08	1	3262
	83	24	4	2021-10-12	1	2755
	90	39	35	2021-10-14	1	842
	92	37	5	2021-10-03	1	95
	109	5	43	2021-10-27	1	942
	111	13	37	2021-11-20	1	531
	112	42	22	2021-12-12	1	717
	114	41	24	2021-12-22	1	1249
	124	45	43	2021-12-26	1	1102
	129	33	15	2021-08-16	1	2139
	131	36	20	2021-07-22	1	2790
	132	26	37	2021-11-06	1	3641
	148	25	18	2022-01-03	1	2487
	150	36	4	2021-12-12	1	989
	168	18	9	2021-11-18	1	2080
	181	36	30	2021-11-14	1	3579
	211	18	30	2021-10-06	1	501
	212	49	15	2021-08-16	1	190
	221	39	23	2021-12-16	1	2564
	229	21	49	2021-10-14	1	2709
	239	7	44	2021-10-15	1	1126
	245	14	2	2021-07-09	1	1941
	255	17	23	2021-11-02	1	661
	277	2	41	2021-09-26	1	2923
	298	12	20	2022-01-07	1	3100
	326	25	15	2022-01-12	1	1833
	331	6	3	2021-07-29	1	3886
	357	25	20	2021-09-11	1	2183
	364	41	37	2021-07-16	1	486
	377	7	34	2021-12-02	1	1213
	388	7	28	2021-12-12	1	886
	406	24	8	2021-07-05	1	3178
	413	46	37	2021-08-19	1	2668
	414	42	1	2022-01-02	1	1267

```
select * from transfusions
inner join hospitals on transfusions.hospitalId = hospitals.id
where hospitals.id != 3;
```

**7. Select patient name and patient id and hospital address, hospital id of patients assigned hospital in descending order by amount of blood transfered in each hospital**

1 • SELECT \* FROM blooddb1.hospitalsandpatientsorderedbybloodranking;

&lt;

Result Grid



Filter Rows:

Export:



Wrap Cell Content:

	id	name	surname	address
▶	3	Veronika	Bosanko	4960 Macpherson Circle
	4	Antonino	Murum	4960 Macpherson Circle
	5	Lynelle	Fitchet	4960 Macpherson Circle
	6	Katha	Tierney	4960 Macpherson Circle
	8	Cristobal	Yoseloff	4960 Macpherson Circle
	10	Mina	Snead	4960 Macpherson Circle
	13	Grantley	Petrosian	4960 Macpherson Circle
	19	Jammie	Gadie	4960 Macpherson Circle
	23	Nari	Grimwad	4960 Macpherson Circle
	29	Miles	Valintine	4960 Macpherson Circle
	32	Winfred	Font	4960 Macpherson Circle
	33	Juana	Antusch	4960 Macpherson Circle
	38	Oswell	England	4960 Macpherson Circle
	42	Prissie	Haacker	4960 Macpherson Circle
	43	Jenn	Harmes	4960 Macpherson Circle
	48	Beau	Cookney	4960 Macpherson Circle
	50	Mariette	Carlton	4960 Macpherson Circle
	20	Lindy	MacMee...	2 Fairview Drive
	22	Keane	Le Count	2 Fairview Drive
	27	Billi	Butting	2 Fairview Drive
	28	Mattie	Snepp	2 Fairview Drive
	41	Othella	Munnery	2 Fairview Drive
	44	Karola	Zini	2 Fairview Drive
	49	Ardeen	Pengelly	2 Fairview Drive
	2	Beverley	Dudgeon	93 Ryan Plaza
	9	Julienne	Pascoe	93 Ryan Plaza
	11	Joete	Wenzel	93 Ryan Plaza
	12	Jake	Ginnell	93 Ryan Plaza
	14	Traver	MacMichael	93 Ryan Plaza
	21	Pavlov	Duester	93 Ryan Plaza
	26	Frankie	Blowen	93 Ryan Plaza
	36	Crosby	Brayford	93 Ryan Plaza
	37	Melly	Grandham	93 Ryan Plaza
	39	Orbadiah	Axford	93 Ryan Plaza
	40	Hestia	Mor	93 Ryan Plaza
	1	Maren	Benford	73567 Grim Junction
	7	Cathlene	Mathon	73567 Grim Junction
	15	Cherin	Clapton	73567 Grim Junction
	16	Gigi	Kaes	73567 Grim Junction
	17	Janifer	Twelftree	73567 Grim Junction
	18	Patty	Glowinski	73567 Grim Junction
	24	Minetta	Rickell	73567 Grim Junction
	25	Shauna	Minihan	73567 Grim Junction
	30	Jenna	Else	73567 Grim Junction
	31	Moselle	Thornton	73567 Grim Junction
	34	Hans	Annice	73567 Grim Junction
	35	Danila	Pacheco	73567 Grim Junction
	45	Dreddy	Curley	73567 Grim Junction
	46	Amalea	Corssen	73567 Grim Junction
	47	Dorie	Craister	73567 Grim Junction

```
select patients.id,patients.name,patients.surname,hospitals.address from patients
inner join hospitals on hospitals.id = patients.hospitalId
order by (Select count(transfusions.bloodTransferredML) from transfusions
inner join hospitals on hospitals.id = transfusions.hospitalId ) desc
```

## 8. Select hospital address, patient name associated with patient of id 23

---



```
select hospitals.address, patients.name from patients inner join hospitals on patients.hospitalId = hospitals.id where patients.id = 23
```

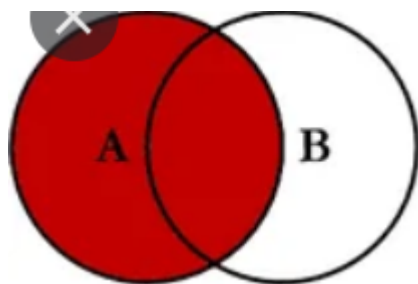
## 9. Select transfusion date, patient name of all patients even if patients didnt had a transfusion (left join example)

---

1 • **SELECT \* FROM** blooddb1.patientstransfusionsleftjoindemo;

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

	date	name	surname
▶	2021-11-23	Maren	Benford
	2021-12-29	Maren	Benford
	2021-10-13	Beverley	Dudgeon
	2021-09-26	Beverley	Dudgeon
	2021-12-02	Beverley	Dudgeon
	2021-07-01	Beverley	Dudgeon
	2022-01-15	Beverley	Dudgeon
	2022-01-09	Beverley	Dudgeon
	2022-01-01	Beverley	Dudgeon
	2021-09-12	Beverley	Dudgeon
	2022-01-13	Veronika	Bosanko
	2021-12-16	Veronika	Bosanko
	2021-10-11	Veronika	Bosanko
	2021-11-17	Veronika	Bosanko
	2021-08-26	Veronika	Bosanko
	2021-12-20	Veronika	Bosanko
	2021-12-21	Veronika	Bosanko
	2021-11-20	Veronika	Bosanko
	2021-09-13	Veronika	Bosanko
	2021-11-09	Veronika	Bosanko
	2021-07-25	Veronika	Bosanko
	2021-10-20	Veronika	Bosanko
	2022-01-10	Veronika	Bosanko
	2021-09-22	Antonino	Murru
	2021-10-20	Antonino	Murru
	2021-10-27	Lynelle	Fitchet
	2022-01-08	Lynelle	Fitchet
	2021-11-27	Lynelle	Fitchet
	2021-08-04	Lynelle	Fitchet
	2021-12-10	Lynelle	Fitchet
	2021-07-21	Lynelle	Fitchet
	2021-07-25	Lynelle	Fitchet
	2021-08-08	Katha	Tierney
	2021-08-11	Katha	Tierney
	2021-12-02	Katha	Tierney
	2021-12-31	Katha	Tierney
	2021-07-17	Katha	Tierney
	2021-07-29	Katha	Tierney
	2021-11-28	Katha	Tierney



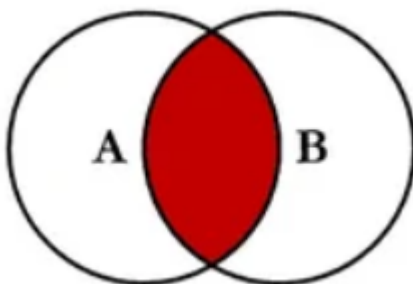
```
select transfusions.date, patients.name,patients.surname from patients
left join transfusions on transfusions.patientId = patients.id
```

**10.Select transfusion date, patient name of all patients that had a transfusion (inner join example)**



1 • `SELECT * FROM blooddb1.patientstransfusionsinnerjoindemo;`

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
date	name	surname	
2021-11-23	Maren	Benford	
2021-12-29	Maren	Benford	
2021-10-13	Beverley	Dudgeon	
2021-09-26	Beverley	Dudgeon	
2021-12-02	Beverley	Dudgeon	
2021-07-01	Beverley	Dudgeon	
2022-01-15	Beverley	Dudgeon	
2022-01-09	Beverley	Dudgeon	
2022-01-01	Beverley	Dudgeon	
2021-09-12	Beverley	Dudgeon	
2022-01-13	Veronika	Bosanko	
2021-12-16	Veronika	Bosanko	
2021-10-11	Veronika	Bosanko	
2021-11-17	Veronika	Bosanko	
2021-08-26	Veronika	Bosanko	
2021-12-20	Veronika	Bosanko	
2021-12-21	Veronika	Bosanko	
2021-11-20	Veronika	Bosanko	
2021-09-13	Veronika	Bosanko	
2021-11-09	Veronika	Bosanko	
2021-07-25	Veronika	Bosanko	
2021-10-20	Veronika	Bosanko	
2022-01-10	Veronika	Bosanko	
2021-09-22	Antonino	Murum	
2021-10-20	Antonino	Murum	
2021-10-27	Lynelle	Fitchet	
2022-01-08	Lynelle	Fitchet	
2021-11-27	Lynelle	Fitchet	
2021-08-04	Lynelle	Fitchet	
2021-12-10	Lynelle	Fitchet	
2021-07-21	Lynelle	Fitchet	



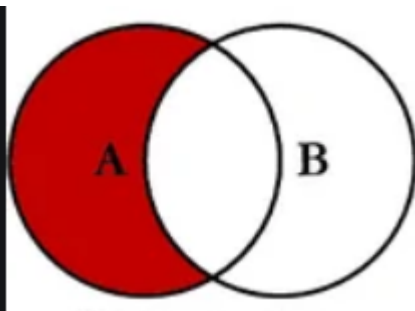
```
select transfusions.date, patients.name,patients.surname from patients
inner join transfusions on transfusions.patientId = patients.id
```

**11. Select transfusion date, patient name of all patients that didnt had a transfusion ( left null join example)**

```
1 • SELECT * FROM blooddb1.patientstransfusionsnulljoindemo;
```

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

date	name	surname
------	------	---------



```
select transfusions.date, patients.name,patients.surname from patients
left join transfusions on transfusions.patientId = patients.id where transfusions.patientId= null
```

## 12. Count how many patients are in every hospital

```
1 • SELECT * FROM blooddb1.hospitalsstats;
```

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

	id	address	postcode	name	district	sum of patients
▶	1	4960 Macpherson Circle	33-115	St. Mary Hospital	West	17
	2	2 Fairview Drive	33-115	Main West Hospital	West	7
	3	93 Ryan Plaza	33-115	Giga Hospital	East	11
	4	73567 Grim Junction	503-1305	St. John Hospital	East	15

```
Select hospitals.*,count(*) as "sum of patients"
from patients inner join hospitals on hospitals.id = patients.hospitalId group by hospitals.id
```

## 13. Count how many patients and donors are assigned to hospital of id 3 (inner join will also work in this example)

1 • `SELECT * FROM blooddb1.countpatientsanddonorsofhospital3;`

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	sum of patients and donors			
▶	24			

```
Select  count( distinct donors.id ) + count( distinct patients.id ) as "sum of patients and donors" from hospitals
left join patients on  patients.hospitalId = hospitals.id
left join donors on   donors.hospitalId = hospitals.id
where hospitals.id = 3
```

**14 Count how many patients and donors are assigned to hospital of id 3 with blood type AB+**

	sum of patients and donors
▶	9

```
Select  count( distinct donors.id ) + count( distinct patients.id ) as "sum of patients and donors" from hospitals
left join patients on  patients.hospitalId = hospitals.id
left join donors on   donors.hospitalId = hospitals.id
where  donors.bloodTypeId = (Select id from blood where blood.fullName='AB+') and patients.bloodTypeId = (Select id from blood where blood.fullName='AB+')
```

## Postgres views

```

create MATERIALIZED view patients_view as
select p.id,p.name,p.surname,p.address,p.postcode,p.telephone_number,
p.needed_blood_ml ,sum(t.blood_transferred_ml) as blood_transferred_ml,
h.name as hospital_name, b.full_name ,h.id as hospital_id,b.id as blood_id
from patients as p
inner join hospitals as h on h.id = p.hospital_id
left join transfusions as t on t.patient_id = p.id --Count the transfusions for each patient even if they are have zero blod_transferred_ml
inner join blood as b on b.id = p.blood_type_id
group by p.id,h.name ,b.full_name,p.name,p.surname,p.address,p.postcode,p.telephone_number,p.needed_blood_ml,h.id,b.id;

create MATERIALIZED view donors_view as
select d.id,d.name,d.surname,d.address,d.telephone_number ,
sum(t.blood_transferred_ml) as blood_transferred_ml, h.name as hospital_name,
b.full_name ,h.id as hospital_id,b.id as blood_id
from donors as d
inner join hospitals as h on h.id = d.hospital_id
left join transfusions as t on t.donor_id = d.id --Count the transfusions for each patient even if they are have zero blod_transferred_ml
inner join blood as b on b.id = d.blood_type_id
group by d.id,h.name ,b.full_name,d.name,d.surname,d.address,d.telephone_number,h.id,b.id;
--every group by possible due to the sum function (aggregate function)

create MATERIALIZED view hospitals_view as
select h.id , h.name, h.address,h.postcode, h.district ,sum(t.blood_transferred_ml) as blood_transferred_ml,
count(distinct p.id) as patients_count, count(distinct d.id) as donors_count
from hospitals as h
left join transfusions as t on t.hospital_id = h.id
left join patients as p on p.hospital_id = h.id
left join donors as d on d.hospital_id = h.id
group by h.id,h.name, h.address,h.postcode, h.district;
--every group by possible due to the sum function (aggregate function)

create MATERIALIZED view transfusions_view as
select t.id, CONCAT (p.name, ' ', p.surname) AS "patient_fullname" ,
bp.full_name as "patient_blood_type_name", CONCAT (d.name, ' ', d.surname) AS "donor_fullname",
bp.full_name as "donor_blood_type_name" ,t.blood_transferred_ml,t.date,p.id as patient_id,d.id as donor_id,h.id as hospital_id
from transfusions as t
inner join hospitals as h on t.hospital_id = h.id
inner join patients as p on t.patient_id = p.id
inner join donors as d on t.donor_id = d.id
inner join blood as bp on bp.id = p.blood_type_id
inner join blood as bd on bd.id = d.blood_type_id
group by t.id,p.name, p.surname,bp.full_name,d.name, d.surname,bp.full_name,t.blood_transferred_ml,t.date,p.id,d.id,h.id;

create MATERIALIZED view blood_view as
select b.id, b.full_name, b.type, b.is_rhesus
from blood as b;

create MATERIALIZED view blood_blood_view as
select bb.id, bb.donor_blood_id, bb.patient_blood_id, bb.is_transferable
from blood_blood as bb;

```

```

CREATE UNIQUE INDEX ON patients_view (id);
CREATE UNIQUE INDEX ON transfusions_view (id);
CREATE UNIQUE INDEX ON hospitals_view (id);
CREATE UNIQUE INDEX ON donors_view (id);
CREATE UNIQUE INDEX ON blood_view (id);
CREATE UNIQUE INDEX ON blood_blood_view (id);

```

## Triggers and functions for refreshing materialized views

```

771 CREATE FUNCTION tg_refresh_patients_view()
772 RETURNS trigger LANGUAGE plpgsql
773 AS $$
774 BEGIN
775     REFRESH MATERIALIZED VIEW CONCURRENTLY patients_view;
776     REFRESH MATERIALIZED VIEW CONCURRENTLY transfusions_view;
777     REFRESH MATERIALIZED VIEW CONCURRENTLY hospitals_view;
778     REFRESH MATERIALIZED VIEW CONCURRENTLY donors_view;
779
780     RETURN NULL;
781 END;$$;
782
783 CREATE FUNCTION tg_refresh_donors_view()

```



```

784 RETURNS trigger LANGUAGE plpgsql AS $$
785 BEGIN
786     REFRESH MATERIALIZED VIEW CONCURRENTLY donors_view;
787     REFRESH MATERIALIZED VIEW CONCURRENTLY transfusions_view;
788     REFRESH MATERIALIZED VIEW CONCURRENTLY hospitals_view;
789     REFRESH MATERIALIZED VIEW CONCURRENTLY patients_view;
790     RETURN NULL;
791 END;$$;
792
793 CREATE FUNCTION tg_refresh_hospitals_view()
794 RETURNS trigger LANGUAGE plpgsql AS $$
795 BEGIN
796     REFRESH MATERIALIZED VIEW CONCURRENTLY hospitals_view;
797     REFRESH MATERIALIZED VIEW CONCURRENTLY transfusions_view;
798     REFRESH MATERIALIZED VIEW CONCURRENTLY donors_view;
799     REFRESH MATERIALIZED VIEW CONCURRENTLY patients_view;
800     RETURN NULL;
801 END;$$;
802
803 CREATE FUNCTION tg_refresh_transfusions_view()
804 RETURNS trigger LANGUAGE plpgsql AS $$
805 BEGIN
806     REFRESH MATERIALIZED VIEW CONCURRENTLY transfusions_view;
807     REFRESH MATERIALIZED VIEW CONCURRENTLY donors_view;
808     REFRESH MATERIALIZED VIEW CONCURRENTLY patients_view;
809     REFRESH MATERIALIZED VIEW CONCURRENTLY hospitals_view;
810     RETURN NULL;
811 END;$$;
812
813 CREATE TRIGGER tg_refresh_patients_view AFTER INSERT OR UPDATE OR DELETE
814 ON patients
815 FOR EACH STATEMENT EXECUTE FUNCTION tg_refresh_patients_view();
816
817 CREATE TRIGGER tg_refresh_hospitals_view AFTER INSERT OR UPDATE OR DELETE
818 ON hospitals
819 FOR EACH STATEMENT EXECUTE FUNCTION tg_refresh_hospitals_view();
820
821 CREATE TRIGGER tg_refresh_transfusions_view AFTER INSERT OR UPDATE OR DELETE
822 ON transfusions
823 FOR EACH STATEMENT EXECUTE FUNCTION tg_refresh_transfusions_view();
824
825 CREATE TRIGGER tg_refresh_donors_view AFTER INSERT OR UPDATE OR DELETE
826 ON donors
827 FOR EACH STATEMENT EXECUTE FUNCTION tg_refresh_donors_view();
828

```

## Procedure to handle correct transfusions

In php code exists similar select to get all potential donors from the same hospital as patient based on blood groups ids and ordered by from the most similar antibodies. Procedure below also updates needed\_blood\_ml value of patient to whom blood was transfused.


```

827
828 CREATE OR REPLACE PROCEDURE createTransfusion(vpatient_id integer, vdonor_id integer, vhospital_id integer, vblood_transferred_ml float)
829     language plpgsql
830     AS $$
831     DECLARE
832         vneeded_blood_ml float;
833     BEGIN
834         SELECT p.needed_blood_ml INTO vneeded_blood_ml FROM patients as p WHERE p.id = vpatient_id;
835
836         IF vblood_transferred_ml > vneeded_blood_ml THEN
837             vneeded_blood_ml := vblood_transferred_ml - vneeded_blood_ml;
838         ELSE
839             vneeded_blood_ml := 0;
840         END IF;
841
842
843         IF ( SELECT COUNT(1)  from donors as d cross join patients as p
844             inner join blood_blood as bb on bb.donor_blood_id = d.blood_type_id and p.blood_type_id = bb.patient_blood_id
845             inner join hospitals as h on h.id = d.hospital_id
846             inner join blood as b on b.id = d.blood_type_id
847             where  bb.is_transferable and p.id = vpatient_id and d.hospital_id = p.hospital_id and vdonor_id = d.id
848             and vhospital_id = d.hospital_id and vpatient_id = p.id and  bb.is_transferable
849             = true)
850         THEN
851             INSERT INTO transfusions(patient_id,donor_id, blood_transferred_ml,hospital_id)
852             VALUES(vpatient_id,vdonor_id,vneeded_blood_ml,vhospital_id);
853
854             UPDATE patients SET needed_blood_ml = vneeded_blood_ml WHERE id = vpatient_id;
855         ELSE
856             RAISE EXCEPTION 'The transfusion is not allowed';
857         END IF;
858     END;$$;

```

# App Views

## Create/Update/Delete Hospital



**BloodDb App**

GENERAL

[Dashboard](#)

OVERVIEW

- Hospitals
- Patients
- Donors
- Transfusions

ABOUT

- About Dashboard
- BloodDb on GitHub

About
GitHub

Hospitals					Order By:	ID	Descending	
ID	Name	Postcode	Address	District	Action	Blood transferred	Patients	Donors
5	GigaHOSpital2	34-433	West	Giga street 34		0	0	<a href="#">Edit</a> <a href="#">Delete</a>
6	Gigahospital2	34-433	West	giga address 2		0	0	<a href="#">Edit</a> <a href="#">Delete</a>
4	St. John Hospital	503-1305	East	73567 Grim Junction	3073365	15	0	<a href="#">Edit</a> <a href="#">Delete</a>
3	Giga Hospital	33-115	East	93 Ryan Plaza	36176294	11	14	<a href="#">Edit</a> <a href="#">Delete</a>
2	Main West Hospital	33-115	West	2 Fairview Drive	12054168	7	12	<a href="#">Edit</a> <a href="#">Delete</a>
1	St. Mary Hospital	33-115	West	4960 Macpherson Circle	54833772	17	19	<a href="#">Edit</a> <a href="#">Delete</a>

[Add Hospital](#)

Name  Address

## Create/Update/Delete Patient

BloodDb App

GENERAL

Dashboard

OVERVIEW

Hospitals

Patients

Donors

Transfusions

ABOUT

About Dashboard

BloodDb on GitHub

AboutGitHub

Patients										Order By:	ID	Descending	
ID	Name	Surname	Blood Type	Address	Postcode	Telephone Number	Hospital	Recieved Blood MI	Needed Blood MI	Action			
50	Mariette	Carlton	AB-	26228 Schurz Pass	9505	577-194-8119	St. Mary Hospital	26198	896	<div><div></div><div></div></div>			
49	Ardeen	Pengelly	B+	0 Shoshone Crossing	9505	953-993-3870	Main West Hospital	27786	333	<div><div></div><div></div></div>			
48	Beau	Cookney	O-	3422 Lakeland Parkway	70120	816-285-6579	St. Mary Hospital	3771	3455	<div><div></div><div></div></div>			
47	Dorie	Craister	O+	797 Helena Trail	18004	874-568-6303	St. John Hospital	9820	0	<div><div></div><div></div></div>			
46	Amalea	Corssen	O-	09319 Norway Maple Court	18004	331-381-6609	St. John Hospital	10137	0	<div><div></div><div></div></div>			
45	Dreddy	Curley	A-	96 Hauk Park	20210	923-107-7819	St. John Hospital	12680	0	<div><div></div><div></div></div>			
44	Karola	Zini	O+	87215 Garrison Crossing	20210	543-727-2776	Main West Hospital	6902	0	<div><div></div><div></div></div>			

## Create/Update/Delete Donor

<div> <div> <div>BloodDb App</div> <div>GENERAL</div> <div>Dashboard</div> <div>OVERVIEW</div> <div>Hospitals</div> <div>Patients</div> <div>Donors</div> <div>Transfusions</div> <div>ABOUT</div> <div>About Dashboard</div> <div>BloodDb on GitHub</div> </div> <div> <div>About</div> <div>GitHub</div> </div> </div>								
Donors								
Order By: ID Descending								
ID	Name	Surname	Blood Type	Address	Telephone Number	Hospital	Donated Blood MI	Action
48	Clemmie	Astling	A+	60822 Glacier Hill Circle	+55 (514) 298-8824	St. Mary Hospital	10758	<a href="#">Edit</a> <a href="#">Delete</a>
47	Gan	Goullee	AB-	58020 Summit Park	+86 (765) 102-4783	Main West Hospital	8635	<a href="#">Edit</a> <a href="#">Delete</a>
46	Bernhard	Adolthine	O+	4653 Talmadge Road	+62 (645) 711-8897	St. Mary Hospital	22236	<a href="#">Edit</a> <a href="#">Delete</a>
45	Alan	Regan	A+	83425 Russell Lane	+385 (370) 556-0228	Main West Hospital	4022	<a href="#">Edit</a> <a href="#">Delete</a>
44	Joane	Hanvey	A-	8 Redwing Lane	+57 (100) 123-8208	Main West Hospital	18602	<a href="#">Edit</a> <a href="#">Delete</a>
43	Merridie	Millthorpe	O-	81 Marquette Lane	+81 (199) 155-9688	Main West Hospital	28945	<a href="#">Edit</a> <a href="#">Delete</a>
42	Filippo	Hucke	AB+	144 Sugar Avenue	+86 (620) 736-5967	Main West Hospital	2479	<a href="#">Edit</a> <a href="#">Delete</a>
41	Margery	Covert	O-	4781 Delladonna Alley	+996 (789) 442-0049	Giga Hospital	46959	<a href="#">Edit</a> <a href="#">Delete</a>

## Create/Update/Delete Transfusion

<div> <div> <div>BloodDb App</div> <div>GENERAL</div> <div>Dashboard</div> <div>OVERVIEW</div> <div>Hospitals</div> <div>Patients</div> <div>Donors</div> <div>Transfusions</div> <div>ABOUT</div> <div>About Dashboard</div> <div>BloodDb on GitHub</div> </div> <div> <div>About</div> <div>GitHub</div> </div> </div>								
Hospitals								
Order By: ID Descending								
ID	Name	Postcode	Address	District	Action	Blood transfered	Patients	Donors
2	Main West Hospital	33-115	West	2 Fairview Drive	4936344	7	12	<a href="#">Edit</a> <a href="#">Delete</a>
3	Giga Hospital	33-115	East	93 Ryan Plaza	15450204	11	14	<a href="#">Edit</a> <a href="#">Delete</a>
4	St. John Hospital	503-1305	East	73567 Grim Junction	1371255	15	0	<a href="#">Edit</a> <a href="#">Delete</a>
6	Gigahospital2	34-433	East	giga address 2		0	0	<a href="#">Edit</a> <a href="#">Delete</a>
7	Giga Hostpital2	234-34	West	giga address 2		0	0	<a href="#">Edit</a> <a href="#">Delete</a>
Add Hospital								
Name	Name			Address	Address			