INSERT INTO donor(d\_id,d\_name,blood\_type,address,contact,email\_address,age)

values(1,'d1','A+','150 NY','93344569','d1@gmail.com',30),

(2,'d2','B-','160 jsq','88816789','d2@gmail.com',32),

(3,'d3','AB+','170 NY','44544569','d3@gmail.com',20),

(4,'d4','AB-','175 NY','93344585','d4@gmail.com',23),

(5,'d5','O+','150 NY','73344569','d5@gmail.com',28),

(6,'d6','O-','150 qeens','93355569','d6@gmail.com',35),

(7,'d7','A+','150 qeens','43344569','d7@gmail.com',30),

(8,'d8','B-','160 jsq','78816789','d8@gmail.com',32),

(9,'d9','AB+','170 NY','54544569','d9@gmail.com',20),

(10,'d10','AB-','175 jsq','13344585','d10@gmail.com',23),

(11,'d11','O+','150 NY','03344569','d11@gmail.com',28),

(12,'d12','B-','150 qeens','23355569','d12@gmail.com',35),

(13,'d13','A+','150 NY','53344569','d13@gmail.com',30),

(14,'d14','B-','160 jsq','82816789','d14@gmail.com',32),

(15,'d15','AB+','170 qeens','42544569','d15@gmail.com',20),

(16,'d16','AB-','175 NY','99344585','d16@gmail.com',23),

(17,'d17','O+','150 NY','73544569','d17@gmail.com',28),

(18,'d18','A-','150 qeens','95355569','d18@gmail.com',35),

(19,'d19','B-','160 jsq','88116789','d19@gmail.com',32),

(20,'d20','A-','170 jsq','44344569','d20@gmail.com',20);

**SELECT \* from donor;**

**Default Contraint:**

**Alter table donor**

**add b\_quantity\_300ml\_bag int default 1;**

SELECT \* from donor;

Graphical user interface

Description automatically generated

INSERT INTO patient(p\_id,p\_name,p\_blood\_type,address,contact,email\_address,age,req\_quantity)

values(1,'p1','A+','160 JSQ','73344569','P1@gmail.com',30,1),

(2,'P2','B+','160 NYC','48816789','P2@gmail.com',32,1),

(3,'P3','O+','170 HOBOKEN','54544569','P3@gmail.com',20,0),

(4,'P4','O-','87 JERSEY CITY','95344585','P4@gmail.com',23,3),

(5,'P5','AB+','150 NYC','74344589','P5@gmail.com',28,0),

(6,'P6','AB-','150 qeens','93355579','P6@gmail.com',35,1),

(7,'p7','B-','160 JSQ','73344569','P7@gmail.com',45,2),

(8,'P8','B+','160 NYC','48816789','P8@gmail.com',65,1),

(9,'P9','O+','170 HOBOKEN','54544569','P9@gmail.com',55,3),

(10,'P10','O-','87 JERSEY CITY','95344585','P10@gmail.com',40,1),

(11,'P11','AB+','150 NYC','74344589','P11@gmail.com',28,2),

(12,'P12','AB-','150 qeens','93355579','P12@gmail.com',35,2);

SELECT \* from patient;

Graphical user interface

Description automatically generated

Create table check\_availability(check\_no int , p\_id int, A\_status varchar(4), primary key(check\_no));

select \* from check\_availability;

ALTER TABLE check\_availability

ADD CONSTRAINT fk\_p\_id FOREIGN KEY (p\_id) REFERENCES patient (p\_id);

INSERT INTO check\_availability(check\_no,p\_id,a\_status)

values(1,1,'Y'),

(2,2,'N'),

(3,4,'Y'),

(4,4,'N'),

(5,4,'N'),

(6,6,'Y'),

(7,6,'Y'),

(8,7,'Y'),

(9,7,'Y'),

(10,8,'N'),

(11,9,'Y'),

(12,9,'Y'),

(13,9,'Y'),

(14,10,'N'),

(15,11,'Y'),

(16,11,'Y'),

(17,12,'Y'),

(18,12,'Y');select \* from check\_availability;

Text

Description automatically generated

INSERT INTO blood\_collection(check\_id,blood\_collector\_name,d\_id,blood\_collector\_id)

values (1,'Tom',1,100),

(2,'AJ',2,110),

(3,'SAM',3,105),

(4,'TOM',4,100),

(5,'AJ',5,110),

(6,'SAM',6,105),

(7,'MIKE',7,80),

(8,'TOM',8,100),

(9,'AJ',9,110),

(10,'SAM',10,105),

(11,'TOM',11,100),

(12,'AJ',12,110),

(13,'SAM',13,105),

(14,'MIKE',14,80),

(15,'MIKE',15,80),

(16,'MIKE',16,80),

(17,'MIKE',17,80),

(18,'AJ',18,110),

(19,'SAM',19,105),

(20,'MIKE',20,80);

A screenshot of a computer

Description automatically generated with medium confidence

Use of **group by** , **order by** statement and aggregation function **count**:

Show the data of how many blood bags taken by each blood collector by using group by on Blood collector id.

: **select blood\_collector\_id, count(d\_id) AS BAGS**

**from blood\_collection**

**group by blood\_collector\_id**

**ORDER BY BAGS DESC ;**

Text

Description automatically generated

**select blood\_type,**

**count(\*) as blood\_count**

**from donor**

**group by blood\_type**

**order by blood\_type;**

Text

Description automatically generated

Use of **Subquery** with **IN** operator and **where** clause

Query:

Display details of patients who has not gotten blood bags.

**select \* from patient**

**where p\_id In (SELECT p\_id from check\_availability**

**where a\_status = 'N');**

A screenshot of a computer

Description automatically generated

Use of **BITWEEN AND** CLAUSE:

Query:

Show the details of the patients who required blood bags between 2 and 3 in ascending order;

select \* from patient

where req\_quantity between 2 and 3

order by req\_quantity;

**A screenshot of a computer

Description automatically generated with medium confidence**

Use of **nested query**, **group by** with **Having** Clause and aggregate function **count**:

Query: Show the blood type that we have given to the more than two patient.

**select p\_blood\_type, count(req\_quantity)**

**from patient**

**where p\_id In (SELECT p\_id from check\_availability**

**where a\_status = 'Y')**

**group by p\_blood\_type having count(req\_quantity)>1;**

A screenshot of a computer

Description automatically generated with medium confidence

View Created on table Donor:

Query : Create a view on Donor table with age of donors are greater than or equal to 30 and display d\_id, d\_name, blood\_type, age.

:

**CREATE VIEW donor\_View AS**

**SELECT d\_id, d\_name, blood\_type, age**

**FROM donor**

**WHERE age >= 30**

**order by age;**

**select \* from donor\_view;**

Text

Description automatically generated

Function Created:

Query: Show all requested blood count from patient table by using Function.

**Create function get\_req\_blood\_quantity(Bags\_from int, Bags\_to int)**

**returns int**

**language plpgsql**

**as**

**$$**

**Declare**

**Bags\_count integer;**

**Begin**

**select count(\*)**

**into Bags\_count**

**from patient**

**where req\_quantity between Bags\_from and Bags\_to;**

**return Bags\_count;**

**End;**

**$$;**

Text

Description automatically generated

Text

Description automatically generated

**Create Procedure:**

Query: Create produre to insert new values in Donor table.

CREATE OR REPLACE PROCEDURE Insert\_new\_donor(In\_d\_id int,

In\_d\_name varchar(30),

In\_blood\_type varchar(30),

In\_address varchar(100),

In\_contact varchar(15),

In\_email\_address varchar(30),

In\_age int,

In\_b\_quantity\_300ml\_bag int)

LANGUAGE plpgsql

AS $$

BEGIN

INSERT INTO donor(d\_id,d\_name,blood\_type,address,contact,email\_address,age, b\_quantity\_300ml\_bag)

values(In\_d\_id,In\_d\_name,In\_blood\_type,In\_address,In\_contact,

In\_email\_address,In\_age, In\_b\_quantity\_300ml\_bag);

--- SQL statements / logic /condition.

END $$

call Insert\_new\_donor(null,'d21','AB+','107 WAVERLY','55132545','d21@gmail.com',);

Procedure Successfully Created:

Text

Description automatically generated

Query Succefully excuted:

Call Insert\_new\_donor(21,'d21','O+','107 waverly','8830994733','d21@gamil.com',25, 1);

Call Insert\_new\_donor(22,'d22','AB+','108 congress','5555994733','d22@gamil.com',18,1);

Text

Description automatically generated

Here is the Output with two new entries in table.

SELECT \* FROM DONOR;

Text

Description automatically generated