DBMS Project Report

Name: Reshma Konepalli (NU ID: 001267356)

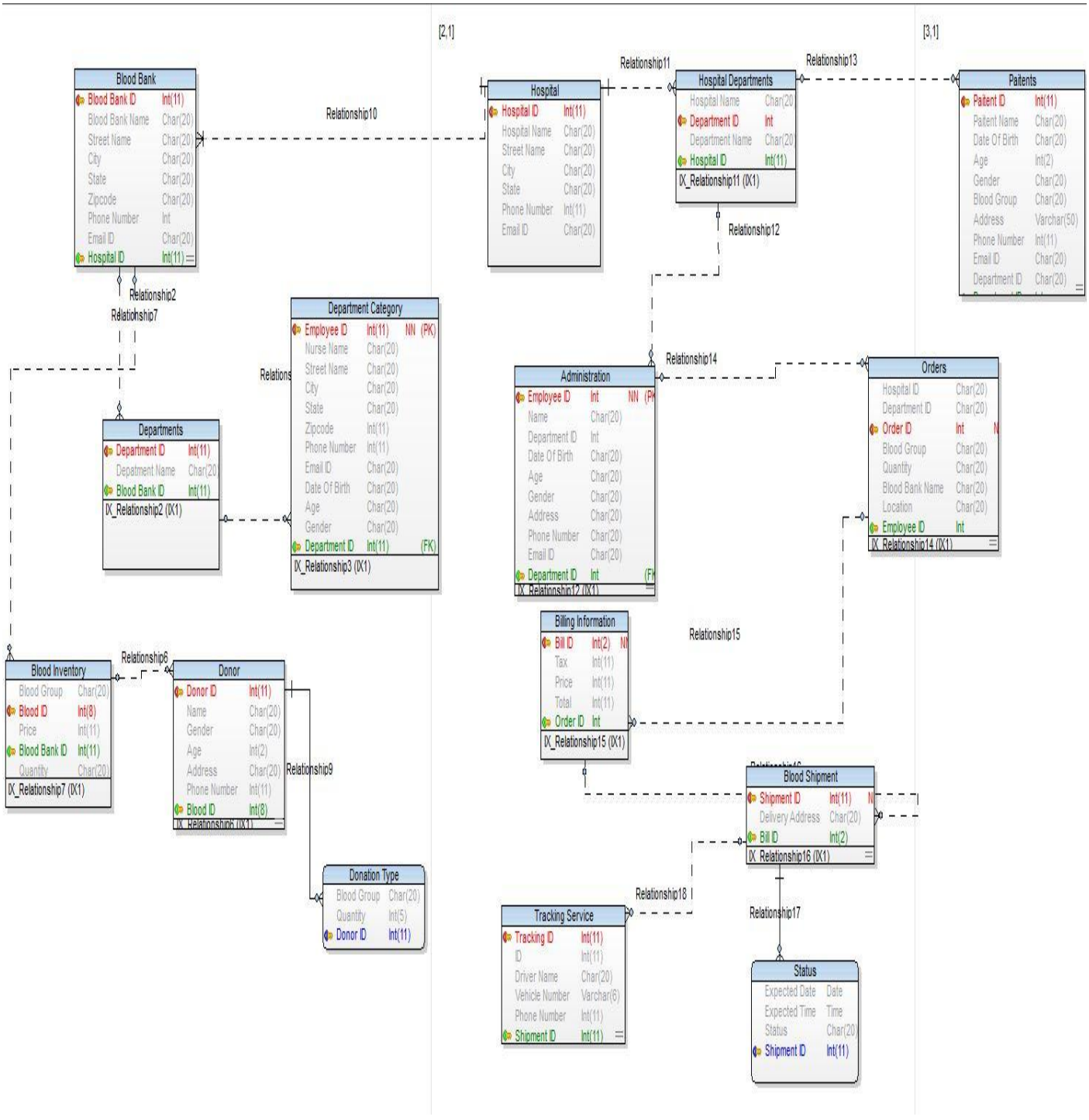
Title:

Blood Donation Management System

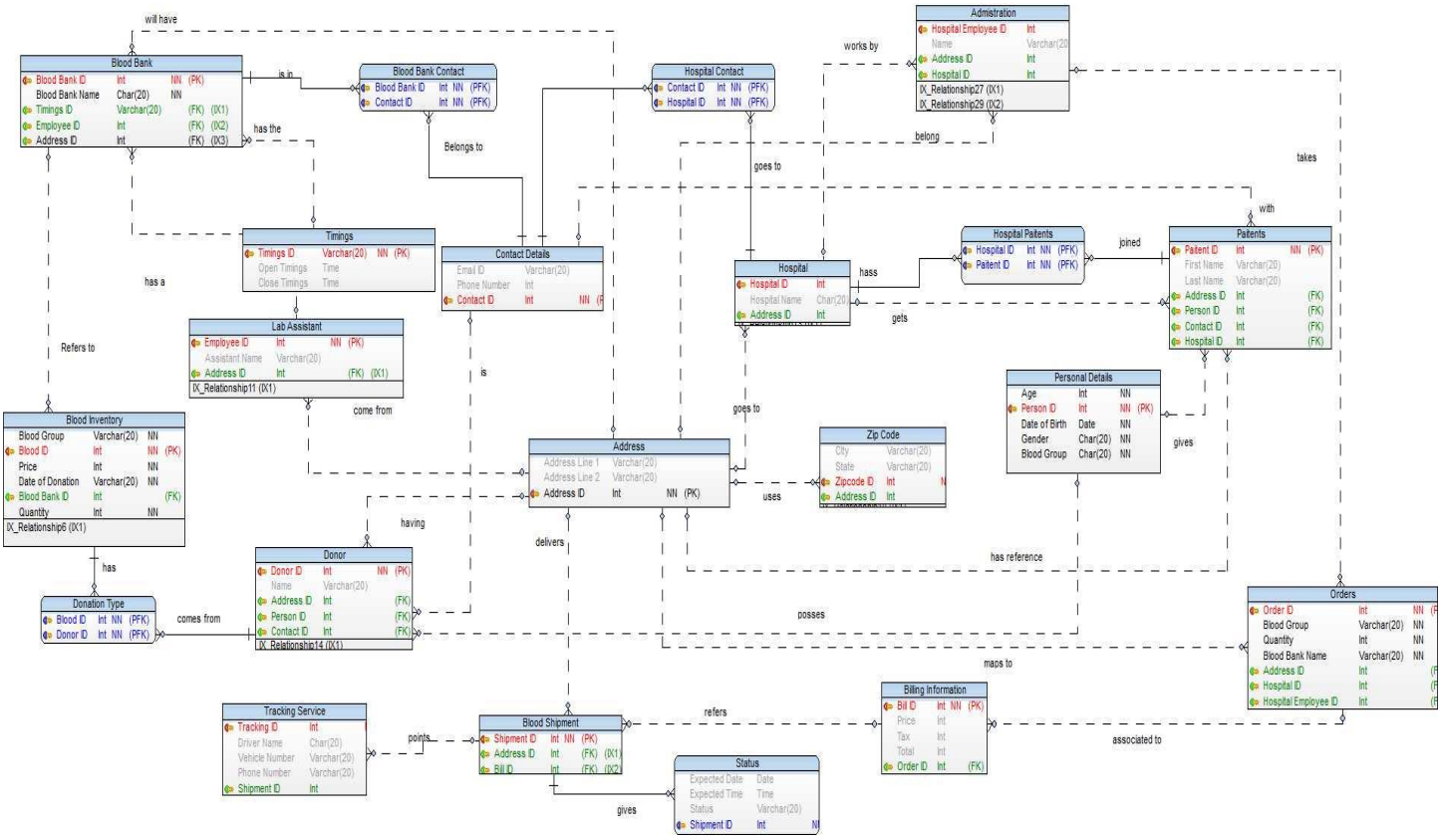
Objective:

This project mainly focuses on Blood Donation Management System which tentatively stores information about the Blood Inventory, data of the patient, data of the donor, employees of the blood bank data of the blood bank, stored blood details, expiry details, orders by hospitals, Medical Facility, Request details, Blood Shipment, Driver details and Tracking Services.

EER Diagram: (Before Normalization)



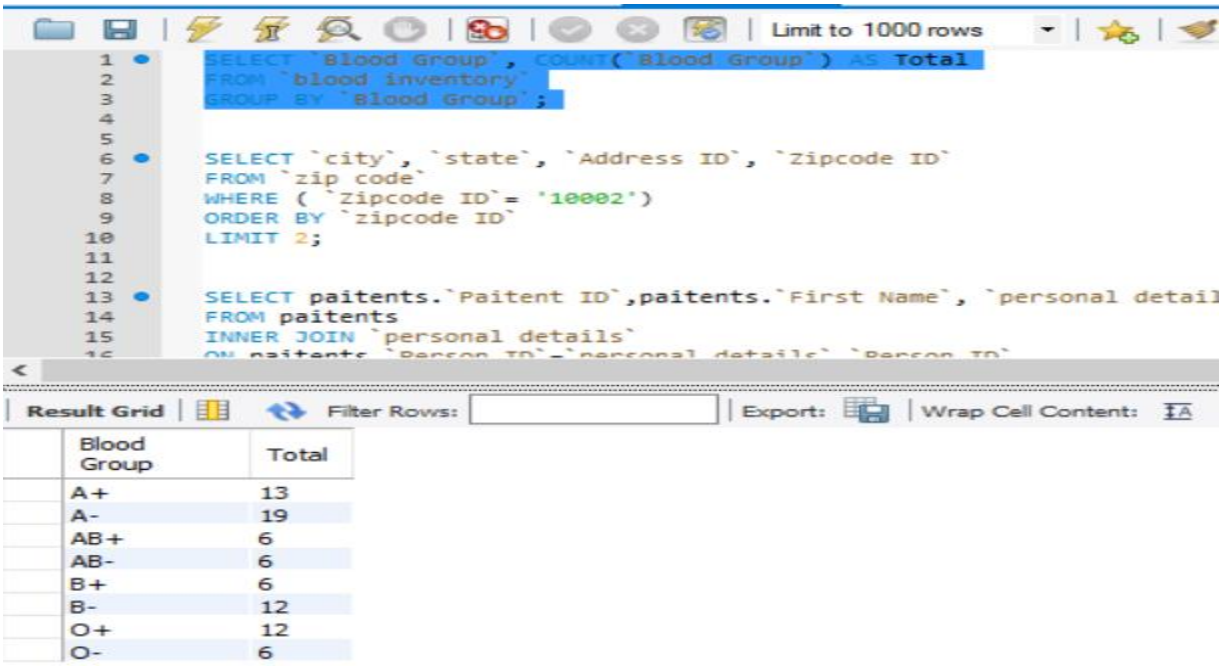
EER Diagram: (After Normalization)



Queries:

1)Find how many bottles of different blood group are stored in blood inventory in a given Blood Bank.

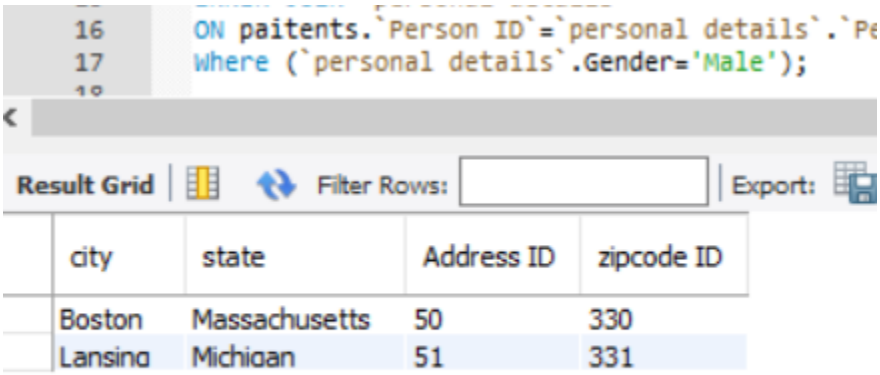
```
SELECT `Blood Group`, COUNT(`Blood Group`) AS Total
FROM `blood inventory`
GROUP BY `Blood Group`;
```



Blood Group	Total
A+	13
A-	19
AB+	6
AB-	6
B+	6
B-	12
O+	12
O-	6

2)Find cities whose name is Boston or their zip code ID is 10002.

```
SELECT `city`, `state`, `Address ID`,`zipcode ID`
FROM `zip code`
WHERE (city= 'Boston' OR `Zipcode ID`= 331)
ORDER BY `zipcode ID`
LIMIT 2;
```



3) Find all the patients who are male and required blood group

```
SELECT paitents.`Paitent ID`,paitents.`First Name`, `personal details`.Age, `personal details`.Gender,
`personal details`.`Blood Group` AS `Required Blood Group`
FROM paitents
INNER JOIN `personal details`
ON paitents.`Person ID`=`personal details`.`Person ID`
WHERE (`personal details`.Gender='Male');
```

```
12
13 •  SELECT paitents.`Paitent ID`,paitents.`First Name`,
14     FROM paitents
```

Result Grid | Filter Rows: | Export: | Wrap

Paitent ID	First Name	Age	Gender	Required Blood Group
661	bbb	53	Male	O-
665	David	58	Male	B-
666	Richard	28	Male	O+
670	Christopher	49	Male	B+
671	Daniel	56	Male	A-
673	Anthony	23	Male	AB+
675	Mark	37	Male	O+
676	Paul	29	Male	A+
677	Steven	24	Male	A-
678	George	21	Male	B-
680	Andrew	57	Male	A-
681	Joshua	45	Male	A+
685	Ronald	43	Male	AB-
686	Timothy	63	Male	AB+

4) To retrieve the number of days a blood group stored since the day of donation.

```
SELECT `blood bank`.`Blood Bank Name`, `contact details`.`Email ID`,
`contact details`.`Phone number`,`blood inventory`.`Blood group`,
datediff(`blood inventory`.`Date of Donation`,'2017/04/20') AS `Days Stored`
From `blood inventory`
INNER JOIN
`blood bank`
ON `blood inventory`.`Blood Bank ID`  = `blood bank`.`Blood Bank ID`
INNER JOIN
`blood bank contact`
ON
`blood bank`.`Blood Bank ID`=`blood bank contact`.`Blood Bank ID`
INNER JOIN
`contact details`
ON
`blood bank contact`.`Contact ID`=`contact details`.`Contact ID`
GROUP BY `blood bank`.`Blood Bank Name`
 LIMIT 10;
```

```
113     ON
114     `blood bank`.`Blood Bank ID`=`blood bank contact`.`Blood Bank ID`
115     INNER JOIN
116     `contact details`
117     ON
```

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

Blood Bank Name	Email ID	Phone number	Blood group	Days Stored
Valley Regional BC	life@company.com	1462106522	O+	NULL
BC of Illinois	life@company.com	1462106522	B-	NULL
BC Ozarks	life@company.com	1500490471	O+	27
Blood Alliance	life@company.com	1462106522	A-	15
Blood Assurance	life@company.com	1462106522	A+	NULL
Community BC	life@company.com	1231161204	AB-	46
Community Blood	life@company.com	1462106522	B+	NULL
Floridas BC	life@company.com	1462106522	A+	11
Heartland BC	life@company.com	1462106522	B-	NULL
Hemacare	life@company.com	1411239218	A+	35

5)Transaction and RollBack

```
START TRANSACTION;
UPDATE `blood inventory` SET `Blood group` = 'B+'
WHERE `Blood ID` = 121;
UPDATE `blood inventory` SET price = '1000'
WHERE `Blood ID` = 123;
SELECT * FROM `blood inventory`;
ROLLBACK;
SELECT * FROM `blood inventory`;
```

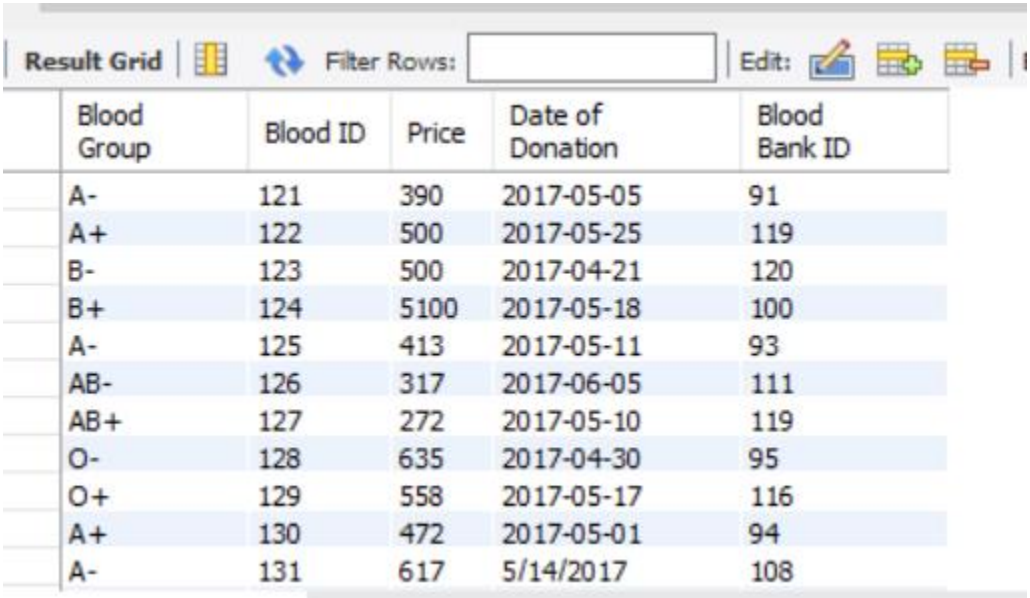
Transaction:

```
122
123     START TRANSACTION;
124     UPDATE `blood inventory` SET `Blood group` = 'B+'
```

Result Grid | Filter Rows: | Edit:

Blood Group	Blood ID	Price	Date of Donation	Blood Bank ID
B+	121	390	2017-05-05	91
A+	122	500	2017-05-25	119
B-	123	1000	2017-04-21	120
B+	124	5100	2017-05-18	100
A-	125	413	2017-05-11	93
AB-	126	317	2017-06-05	111
AB+	127	272	2017-05-10	119
O-	128	635	2017-04-30	95
O+	129	558	2017-05-17	116
A+	130	472	2017-05-01	94
A-	131	617	5/14/2017	108

After RollBack :



Result Grid | Filter Rows: | Edit:

Blood Group	Blood ID	Price	Date of Donation	Blood Bank ID
A-	121	390	2017-05-05	91
A+	122	500	2017-05-25	119
B-	123	500	2017-04-21	120
B+	124	5100	2017-05-18	100
A-	125	413	2017-05-11	93
AB-	126	317	2017-06-05	111
AB+	127	272	2017-05-10	119
O-	128	635	2017-04-30	95
O+	129	558	2017-05-17	116
A+	130	472	2017-05-01	94
A-	131	617	5/14/2017	108

Triggers:

1)Updating the patients First Name and also recording the date and time at which the name has been Updated.

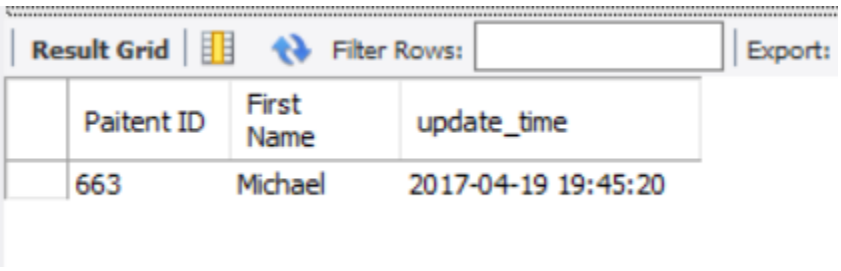
```
create table
UpdatePaitentDetail
(`Paitent ID` int, `First Name` varchar(20), update_time Datetime )

delimiter \\
create trigger UpdatePaitentDetail_trigger
after update on paitents
for each row
begin
declare new_date datetime;
set new_date=now();

insert into UpdatePaitentDetail(`Paitent ID`,`First Name`,update_time)
values(old.`Paitent ID`,old.`First Name`, new_date);

end\\
-----test-----
update paitents set `First Name`='sam'
where `Paitent ID`=663;

select * from UpdatePaitentDetail;
```



Result Grid | Filter Rows: | Export:

Paitent ID	First Name	update_time
663	Michael	2017-04-19 19:45:20

2)Updating the price per bottle of a particular blood group
Create table

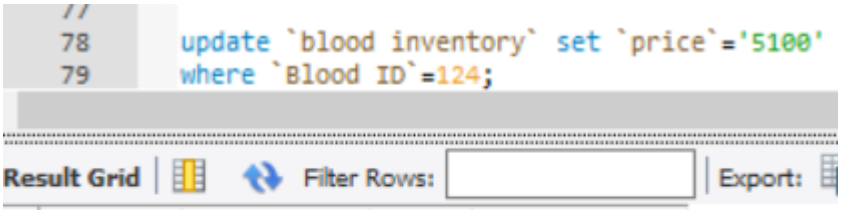
UpdatedPrice
(`Blood ID` int, `Blood Group` Varchar(20), Price int, updated_price int)

```
delimiter \\
create trigger UpdatedPrice_trigger
after update on `blood inventory`
for each row
begin
declare updated_price int;
set updated_price= new.price;

insert into UpdatedPrice(`Blood ID`, `Blood Group`, Price, updated_price)
values(old.`Blood ID`,old.`Blood Group`,old.Price, updated_price);

end \\

-----test-----
select * from UpdatedPrice;
update `blood inventory` set `price`='500'
where `Blood ID`=122;
update `blood inventory` set `price`='500'
where `Blood ID`=123;
update `blood inventory` set `price`='5100'
where `Blood ID`=124;
```



Blood ID	Blood Group	Price	updated_price
122	A+	500	500
123	B-	532	500
124	B+	719	5100

3) Given a Bill information this trigger sums up the price and the tax resulting total bill.
CREATE TABLE Total_Bill (price int, Tax DECIMAL(10,2));

```
 CREATE TRIGGER TotalBill_tigger
 BEFORE INSERT ON Total_Bill

 FOR EACH ROW

 SET @sum = new.price + new.tax;
 SET @sum=0;
 INSERT INTO Total_Bill(price, Tax)
 VALUES(1370,14.98);
SELECT @sum AS 'Total amount';
```

30

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

Total amount
1384.98

Stored Procedures:

1) Maximum Quantity of Order received for each blood group.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `MaxQuantityOrder_BloodGroup`()
BEGIN
SELECT admistration.`name` as `Ordered person name`,orders.`order ID`, orders.`Blood Group`, MAX(
orders.`Quantity`), orders.`blood bank name`
FROM
   orders
INNER JOIN
   admistration
 ON
 orders.`hospital employee ID`=admistration.`hospital employee ID`

GROUP BY  orders.`Blood Group`;

END
```

-----test-----

-→ call MaxQuantityOrder _BloodGroup ();

```
45  drop trigger UpdatePaitentDetail_trigger;
46  call GetMostDemanded_BloodGroup ();
47
48  call GetCommonAge_Donor();
49
```

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

Ordered person name	order ID	Blood Group	MAX(orders.`Quantity`)	blood bank name
Viacheslav Benchimol	702	A+	16	Blood Alliance
Nicole Gao	701	A-	20	OneBlood
Yan Austin	707	AB+	17	Northwest Florida BC
Thanadtha Brown	706	AB-	17	LifeSouth BC
William Barrett	704	B+	16	OneBlood
Scott Cheng	703	B-	20	Community Blood
Matthew Hanan	709	O+	19	Southeastern BC
Montero Greene	708	O-	18	United States BC

2) To get the most common age of the Blood Donors.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `GetCommonAge_Donor`()
BEGIN
SELECT Donor.`Donor ID`,Donor.`name` as `Donor Name`, `personal details`.`Age`, `personal
details`.`Gender`
```

```
FROM
   Donor
INNER JOIN
   `personal details`
ON  Donor.`Person ID`=`personal details`.`Person ID`

WHERE `personal details`.`Age` between 22 and 35

GROUP BY  `personal details`.`Age`;
END
----test-----
Call GetCommonAge_Donor();
```

```
48    call GetCommonAge_Donor();
49
50    call MaxOpenHours();
51
52    SELECT * FROM `orderstatus`;
53
54
```

Result Grid | Filter Rows: | Export:

Donor ID	Donor Name	Age	Gender
576	Spencer Chang	24	Male
574	Taylor Buckley	30	Female
575	Sung Casasavas	31	Male
578	Sativa Cheung	32	Female
585	Owen Franzoni	35	Male

3) To Display all the Blood Banks that are open for more than 12 hours.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `MaxOpenHours`()
BEGIN
SELECT  `Blood Bank`.`Blood Bank Name`,TIMEDIFF(Timings.`close timings`,Timings.`Open timings`) As
`Maximum open Hours`,
 `address`.`address line 1`,`address`.`address line 2`,`Zip Code`.`city`, `Zip Code`.`State`,`Zip Code`.`Zipcode
ID`
FROM
   `blood bank`
     INNER JOIN
      timings
      ON  `blood bank`.`Timings ID`=timings.`Timings ID`
     INNER JOIN
      address
      ON `blood bank`.`Address ID`= address.`Address ID`
      INNER JOIN
      `Zip Code`
      ON address.`Address ID`=`Zip Code`.`Address ID`

  WHERE Timediff(Timings.`close timings`,Timings.`Open timings`)> '12:00:00'

 GROUP BY `Blood Bank`.`Blood Bank Name`;
END

----test-----
Call MaxOpenHours()
```

```
44
drop trigger UpdatePaitentDetail_trigger;
```

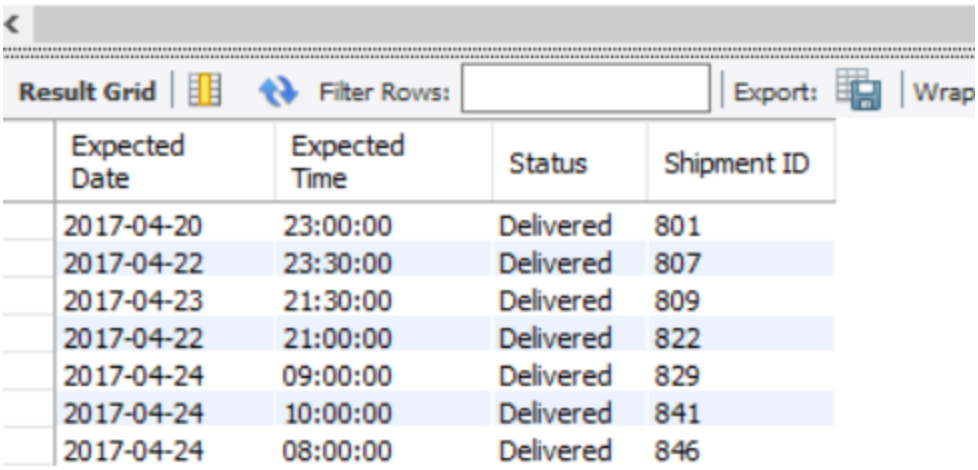
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Blood Bank Name	Maximum open Hours	address line 1	address line 2	city	State	Zipcode ID
BC of Illinois	13:00:00	129	pittsburg	Frankfort	Kentucky	326
BC Ozarks	14:30:00	796	kassone st	Lincoln	Nebraska	336
Blood Alliance	15:00:00	313	Huntiaton Avenue	Juneau	Alaska	311
Blood Assurance	12:30:00	166	private alley	Boise	Idaho	321
Community BC	12:45:00	439	arliaelly	Lansing	Michigan	331
Community Blood	14:30:00	908	Jamaica Plain	Phoenix	Arizona	312
Floridas BC	14:15:00	818	tetlow	Sacramento	California	314
Heartland BC	13:00:00	542	peterborough	Baton Rouge	Louisiana	327
Hemacare	14:00:00	778	maxie road	Trenton	New Jersey	339
Indiana BC	15:00:00	426	groove st	Jackson	Mississippi	333
LifeSouth BC	14:30:00	342	framigham	Denver	Colorado	315
Northwest Florid...	15:00:00	528	roxbury	Hartford	Connecticut	316
NY Blood Center	14:00:00	442	soullivan st	Santa Fe	New Mexico	340
OneBlood	13:00:00	347	bolyston street	Little Rock	Arkansas	313
Shepeard BC	12:30:00	566	harmony	Indianapolis	Indiana	323
South Bend Med...	16:30:00	304	droveeing st	Jefferson City	Missouri	334
Southeastern BB	13:30:00	122	jhonsson	Des Moines	Iowa	324
Suncoast Blood ...	12:15:00	915	hammond street	Atlanta	Georgia	319
United States BC	13:30:00	522	symphony	Dover	Delaware	317

Views:

1) To view details of all the delivered orders.

```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `orderstatus` AS
    (SELECT
        `status`.`Expected Date` AS `Expected Date`,
        `status`.`Expected Time` AS `Expected Time`,
        `status`.`Status` AS `Status`,
        `status`.`Shipment ID` AS `Shipment ID`
    FROM
        `status`
    WHERE
        ((`status`.`Status` = 'Delivered')
            AND (CAST(`status`.`Expected Date` AS DATE) < '2017-04-25')));
```

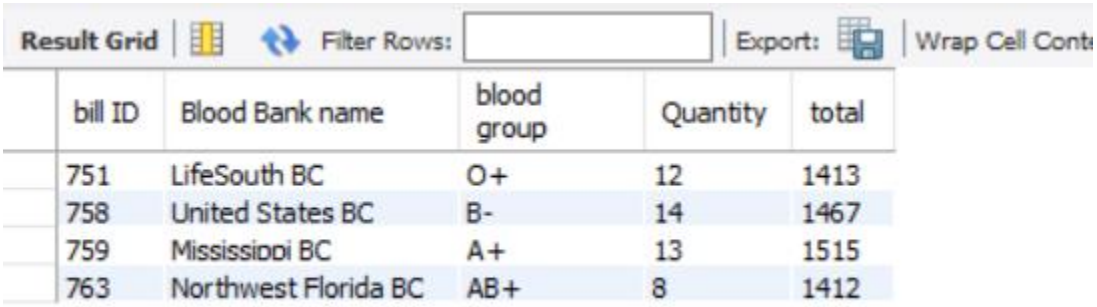


Result Grid | Filter Rows: | Export: | Wrap

Expected Date	Expected Time	Status	Shipment ID
2017-04-20	23:00:00	Delivered	801
2017-04-22	23:30:00	Delivered	807
2017-04-23	21:30:00	Delivered	809
2017-04-22	21:00:00	Delivered	822
2017-04-24	09:00:00	Delivered	829
2017-04-24	10:00:00	Delivered	841
2017-04-24	08:00:00	Delivered	846

2) To view the details of orders whose total bill is greater than 1400.

```
CREATE
    ALGORITHM = UNDEFINED
    DEFINER = `root`@`localhost`
    SQL SECURITY DEFINER
VIEW `topfiveorders` AS
    (SELECT DISTINCT
        `billing information`.`Bill ID` AS `bill ID`,
        `orders`.`Blood Bank Name` AS `Blood Bank name`,
        `orders`.`Blood Group` AS `blood group`,
        `orders`.`Quantity` AS `Quantity`,
        `billing information`.`Total` AS `total`
    FROM
        (`orders`
        JOIN `billing information` ON ((`orders`.`Order ID` = `billing information`.`Order ID`)))
    WHERE
        (`billing information`.`Total` > 1400)
    LIMIT 4)
```



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

bill ID	Blood Bank name	blood group	Quantity	total
751	LifeSouth BC	O+	12	1413
758	United States BC	B-	14	1467
759	Mississippi BC	A+	13	1515
763	Northwest Florida BC	AB+	8	1412

Blood Bank Database Final Dump:

```
/*
Created: 4/15/2017
Modified: 4/19/2017
Model: MySQL 5.5
Database: MySQL 5.5
*/


-- Create tables section -------------------------------------------------
```

```
-- Table Blood Bank

CREATE TABLE `Blood Bank`
(
  `Blood Bank ID` Int NOT NULL,
  `Blood Bank Name` Char(20) NOT NULL,
  `Timings ID` Varchar(20),
  `Employee ID` Int,
  `Address ID` Int
)
;

CREATE INDEX `IX_Relationship1` ON `Blood Bank` (`Timings ID`)
;

CREATE INDEX `IX_Relationship9` ON `Blood Bank` (`Employee ID`)
;

CREATE INDEX `IX_Relationship12` ON `Blood Bank` (`Address ID`)
;

ALTER TABLE `Blood Bank` ADD  PRIMARY KEY (`Blood Bank ID`)
;

-- Table Blood Inventory

CREATE TABLE `Blood Inventory`
(
  `Blood Group` Varchar(20) NOT NULL,
  `Blood ID` Int NOT NULL,
  `Price` Int NOT NULL,
  `Date of Donation` Varchar(20) NOT NULL,
  `Blood Bank ID` Int
)
;

CREATE INDEX `IX_Relationship6` ON `Blood Inventory` (`Blood Bank ID`)
;

ALTER TABLE `Blood Inventory` ADD  PRIMARY KEY (`Blood ID`)
;

-- Table Lab Assistant

CREATE TABLE `Lab Assistant`
(
  `Employee ID` Int NOT NULL,
  `Assistant Name` Varchar(20),
  `Address ID` Int
)
;

CREATE INDEX `IX_Relationship11` ON `Lab Assistant` (`Address ID`)
;
```

```
ALTER TABLE `Lab Assistant` ADD  PRIMARY KEY (`Employee ID`)
;

-- Table Donor

CREATE TABLE `Donor`
(
  `Donor ID` Int NOT NULL,
  `Name` Varchar(20),
  `Address ID` Int,
  `Person ID` Int,
  `Contact ID` Int
)
;

CREATE INDEX `IX_Relationship14` ON `Donor` (`Address ID`)
;

CREATE INDEX `IX_Relationship24` ON `Donor` (`Person ID`)
;

CREATE INDEX `IX_Relationship25` ON `Donor` (`Contact ID`)
;

ALTER TABLE `Donor` ADD  PRIMARY KEY (`Donor ID`)
;

-- Table Timings

CREATE TABLE `Timings`
(
  `Timings ID` Varchar(20) NOT NULL,
  `Open Timings` Time,
  `Close Timings` Time
)
;

ALTER TABLE `Timings` ADD  PRIMARY KEY (`Timings ID`)
;

-- Table Contact Details

CREATE TABLE `Contact Details`
(
  `Email ID` Varchar(20),
  `Phone Number` Int,
  `Contact ID` Int NOT NULL
)
;

ALTER TABLE `Contact Details` ADD  PRIMARY KEY (`Contact ID`)
;

-- Table Address

CREATE TABLE `Address`
```

```
(
  `Address Line 1` Varchar(20),
  `Address Line 2` Varchar(20),
  `Address ID` Int NOT NULL
)
;

ALTER TABLE `Address` ADD  PRIMARY KEY (`Address ID`)
;

-- Table Zip Code

CREATE TABLE `Zip Code`
(
  `CIty` Varchar(20),
  `State` Varchar(20),
  `Zipcode ID` Int NOT NULL,
  `Address ID` Int
)
;

CREATE INDEX `IX_Relationship10` ON `Zip Code` (`Address ID`)
;

ALTER TABLE `Zip Code` ADD  PRIMARY KEY (`Zipcode ID`)
;

-- Table Hospital

CREATE TABLE `Hospital`
(
  `Hospital ID` Int NOT NULL,
  `Hospital Name` Char(20),
  `Address ID` Int
)
;

CREATE INDEX `IX_Relationship13` ON `Hospital` (`Address ID`)
;

ALTER TABLE `Hospital` ADD  PRIMARY KEY (`Hospital ID`)
;

-- Table Admistration

CREATE TABLE `Admistration`
(
  `Hospital Employee ID` Int NOT NULL,
  `Name` Varchar(20),
  `Address ID` Int,
  `Hospital ID` Int
)
;

CREATE INDEX `IX_Relationship27` ON `Admistration` (`Address ID`)
;
```

```
CREATE INDEX `IX_Relationship29` ON `Admistration` (`Hospital ID`)
;

ALTER TABLE `Admistration` ADD  PRIMARY KEY (`Hospital Employee ID`)
;

-- Table Paitents

CREATE TABLE `Paitents`
(
  `Paitent ID` Int NOT NULL,
  `First Name` Varchar(20),
  `Last Name` Varchar(20),
  `Address ID` Int,
  `Person ID` Int,
  `Contact ID` Int
)
;

CREATE INDEX `IX_Relationship15` ON `Paitents` (`Address ID`)
;

CREATE INDEX `IX_Relationship23` ON `Paitents` (`Person ID`)
;

CREATE INDEX `IX_Relationship26` ON `Paitents` (`Contact ID`)
;

ALTER TABLE `Paitents` ADD  PRIMARY KEY (`Paitent ID`)
;

-- Table Orders

CREATE TABLE `Orders`
(
  `Order ID` Int NOT NULL,
  `Blood Group` Varchar(20) NOT NULL,
  `Quantity` Int NOT NULL,
  `Blood Bank Name` Varchar(20) NOT NULL,
  `Address ID` Int,
  `Hospital ID` Int,
  `Hospital Employee ID` Int
)
;

CREATE INDEX `IX_Relationship16` ON `Orders` (`Address ID`)
;

CREATE INDEX `IX_Relationship28` ON `Orders` (`Hospital ID`)
;

CREATE INDEX `IX_Relationship30` ON `Orders` (`Hospital Employee ID`)
;

ALTER TABLE `Orders` ADD  PRIMARY KEY (`Order ID`)
```

```
;

-- Table Billing Information

CREATE TABLE `Billing Information`
(
  `Bill ID` Int NOT NULL,
  `Price` Int,
  `Tax` Int,
  `Total` Int,
  `Order ID` Int
)
;

CREATE INDEX `IX_Relationship21` ON `Billing Information` (`Order ID`)
;

ALTER TABLE `Billing Information` ADD  PRIMARY KEY (`Bill ID`)
;

-- Table Blood Shipment

CREATE TABLE `Blood Shipment`
(
  `Shipment ID` Int NOT NULL,
  `Address ID` Int,
  `Bill ID` Int
)
;

CREATE INDEX `IX_Relationship18` ON `Blood Shipment` (`Address ID`)
;

CREATE INDEX `IX_Relationship22` ON `Blood Shipment` (`Bill ID`)
;

ALTER TABLE `Blood Shipment` ADD  PRIMARY KEY (`Shipment ID`)
;

-- Table Tracking Service

CREATE TABLE `Tracking Service`
(
  `Tracking ID` Int NOT NULL,
  `Driver Name` Char(20),
  `Vehicle Number` Varchar(20),
  `Phone Number` Varchar(20),
  `Shipment ID` Int
)
;

CREATE INDEX `IX_Relationship19` ON `Tracking Service` (`Shipment ID`)
;

ALTER TABLE `Tracking Service` ADD  PRIMARY KEY (`Tracking ID`)
;
```

```
-- Table Status

CREATE TABLE `Status`
(
  `Expected Date` Date,
  `Expected Time` Time,
  `Status` Varchar(20),
  `Shipment ID` Int NOT NULL
)
;

ALTER TABLE `Status` ADD  PRIMARY KEY (`Shipment ID`)
;

-- Table Personal Details

CREATE TABLE `Personal Details`
(
  `Age` Int NOT NULL,
  `Person ID` Int NOT NULL,
  `Date of Birth` Date NOT NULL,
  `Gender` Char(20) NOT NULL,
  `Blood Group` Char(20) NOT NULL
)
;

ALTER TABLE `Personal Details` ADD  PRIMARY KEY (`Person ID`)
;

-- Table Blood Bank Contact

CREATE TABLE `Blood Bank Contact`
(
  `Blood Bank ID` Int NOT NULL,
  `Contact ID` Int NOT NULL
)
;

ALTER TABLE `Blood Bank Contact` ADD  PRIMARY KEY (`Blood Bank ID`,`Contact ID`)
;

-- Table Hospital Contact

CREATE TABLE `Hospital Contact`
(
  `Contact ID` Int NOT NULL,
  `Hospital ID` Int NOT NULL
)
;

ALTER TABLE `Hospital Contact` ADD  PRIMARY KEY (`Contact ID`,`Hospital ID`)
;

-- Table Donation Type
```

```
CREATE TABLE `Donation Type`
(
  `Blood ID` Int NOT NULL,
  `Donor ID` Int NOT NULL
)
;

ALTER TABLE `Donation Type` ADD  PRIMARY KEY (`Blood ID`,`Donor ID`)
;

-- Create relationships section -----------------------------------------------

ALTER TABLE `Blood Bank` ADD CONSTRAINT `Relationship1` FOREIGN KEY (`Timings ID`)
REFERENCES `Timings` (`Timings ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Blood Bank Contact` ADD CONSTRAINT `Relationship2` FOREIGN KEY (`Blood Bank ID`)
REFERENCES `Blood Bank` (`Blood Bank ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Blood Bank Contact` ADD CONSTRAINT `Relationship3` FOREIGN KEY (`Contact ID`)
REFERENCES `Contact Details` (`Contact ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Hospital Contact` ADD CONSTRAINT `Relationship4` FOREIGN KEY (`Contact ID`)
REFERENCES `Contact Details` (`Contact ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Hospital Contact` ADD CONSTRAINT `Relationship5` FOREIGN KEY (`Hospital ID`)
REFERENCES `Hospital` (`Hospital ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Blood Inventory` ADD CONSTRAINT `Relationship6` FOREIGN KEY (`Blood Bank ID`)
REFERENCES `Blood Bank` (`Blood Bank ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Donation Type` ADD CONSTRAINT `Relationship7` FOREIGN KEY (`Blood ID`)
REFERENCES `Blood Inventory` (`Blood ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Donation Type` ADD CONSTRAINT `Relationship8` FOREIGN KEY (`Donor ID`)
REFERENCES `Donor` (`Donor ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Blood Bank` ADD CONSTRAINT `Relationship9` FOREIGN KEY (`Employee ID`)
REFERENCES `Lab Assistant` (`Employee ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Zip Code` ADD CONSTRAINT `Relationship10` FOREIGN KEY (`Address ID`)
REFERENCES `Address` (`Address ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Lab Assistant` ADD CONSTRAINT `Relationship11` FOREIGN KEY (`Address ID`)
REFERENCES `Address` (`Address ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;
```

```
ALTER TABLE `Blood Bank` ADD CONSTRAINT `Relationship12` FOREIGN KEY (`Address ID`)
REFERENCES `Address` (`Address ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Hospital` ADD CONSTRAINT `Relationship13` FOREIGN KEY (`Address ID`) REFERENCES
`Address` (`Address ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Donor` ADD CONSTRAINT `Relationship14` FOREIGN KEY (`Address ID`) REFERENCES
`Address` (`Address ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Paitents` ADD CONSTRAINT `Relationship15` FOREIGN KEY (`Address ID`) REFERENCES
`Address` (`Address ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Orders` ADD CONSTRAINT `Relationship16` FOREIGN KEY (`Address ID`) REFERENCES
`Address` (`Address ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Blood Shipment` ADD CONSTRAINT `Relationship18` FOREIGN KEY (`Address ID`)
REFERENCES `Address` (`Address ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Tracking Service` ADD CONSTRAINT `Relationship19` FOREIGN KEY (`Shipment ID`)
REFERENCES `Blood Shipment` (`Shipment ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Status` ADD CONSTRAINT `Relationship20` FOREIGN KEY (`Shipment ID`) REFERENCES
`Blood Shipment` (`Shipment ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Billing Information` ADD CONSTRAINT `Relationship21` FOREIGN KEY (`Order ID`)
REFERENCES `Orders` (`Order ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Blood Shipment` ADD CONSTRAINT `Relationship22` FOREIGN KEY (`Bill ID`)
REFERENCES `Billing Information` (`Bill ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Paitents` ADD CONSTRAINT `Relationship23` FOREIGN KEY (`Person ID`) REFERENCES
`Personal Details` (`Person ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Donor` ADD CONSTRAINT `Relationship24` FOREIGN KEY (`Person ID`) REFERENCES
`Personal Details` (`Person ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Donor` ADD CONSTRAINT `Relationship25` FOREIGN KEY (`Contact ID`) REFERENCES
`Contact Details` (`Contact ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Paitents` ADD CONSTRAINT `Relationship26` FOREIGN KEY (`Contact ID`) REFERENCES
`Contact Details` (`Contact ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;
```

```
ALTER TABLE `Admistration` ADD CONSTRAINT `Relationship27` FOREIGN KEY (`Address ID`)
REFERENCES `Address` (`Address ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Orders` ADD CONSTRAINT `Relationship28` FOREIGN KEY (`Hospital ID`) REFERENCES
`Hospital` (`Hospital ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Admistration` ADD CONSTRAINT `Relationship29` FOREIGN KEY (`Hospital ID`)
REFERENCES `Hospital` (`Hospital ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;

ALTER TABLE `Orders` ADD CONSTRAINT `Relationship30` FOREIGN KEY (`Hospital Employee ID`)
REFERENCES `Admistration` (`Hospital Employee ID`) ON DELETE RESTRICT ON UPDATE RESTRICT
;
```