**DBMS Project Report**

Name: Reshma Konepalli (NU ID: 001267356)

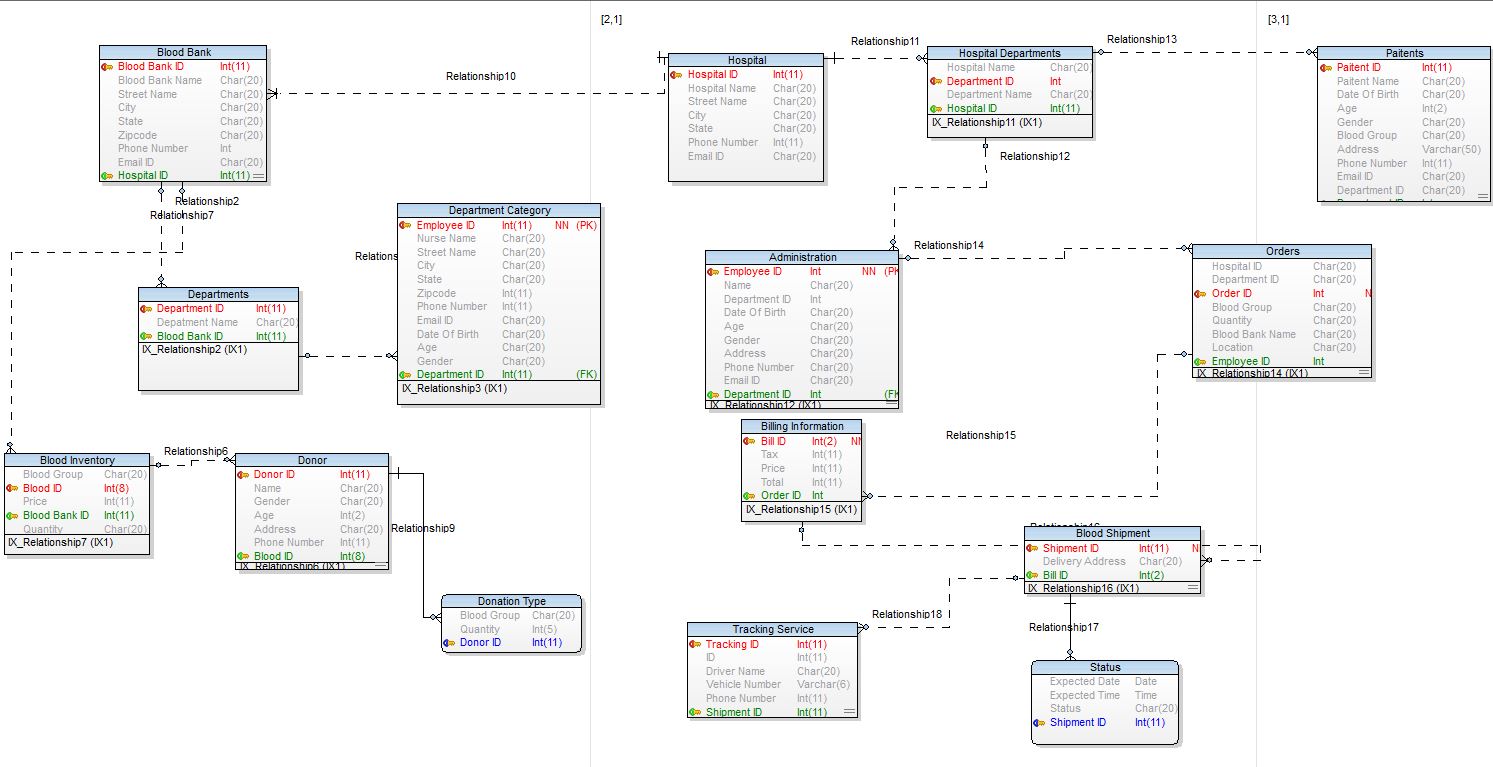
**Title:**

Blood Donation Management System

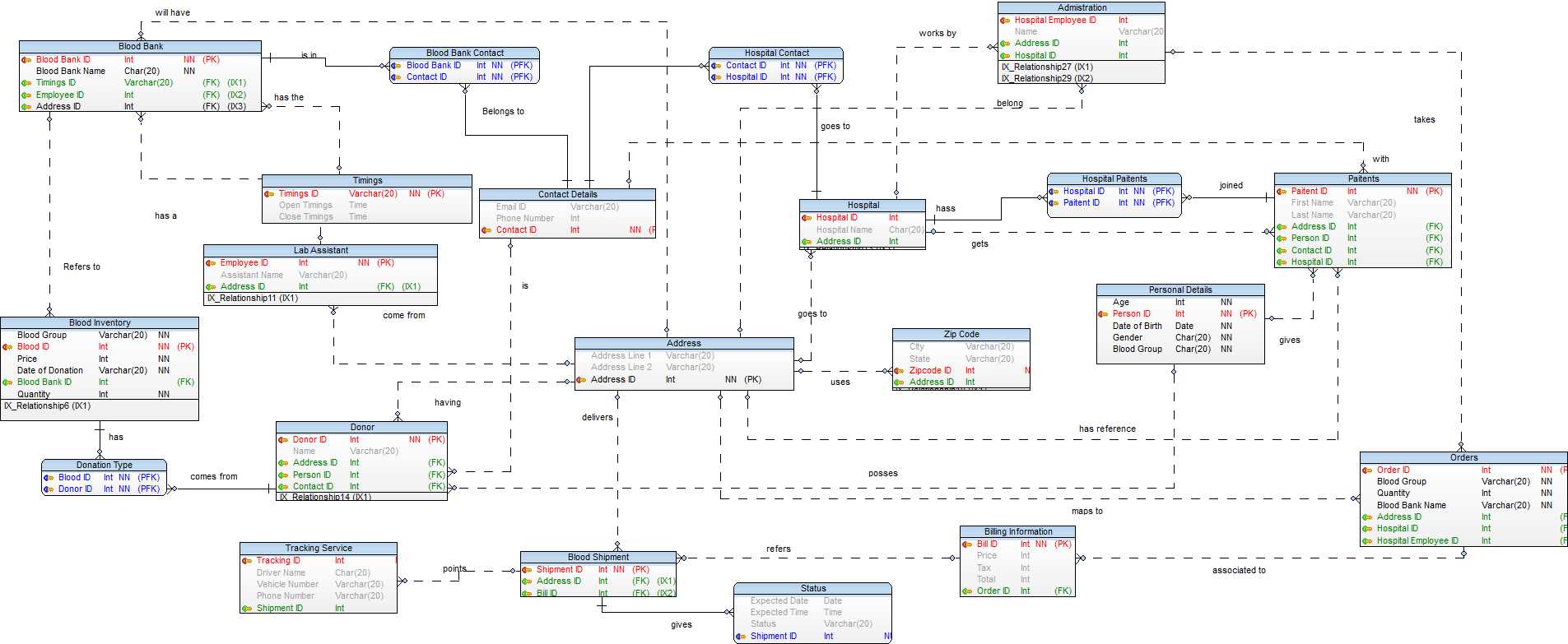
**Objective:**

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)



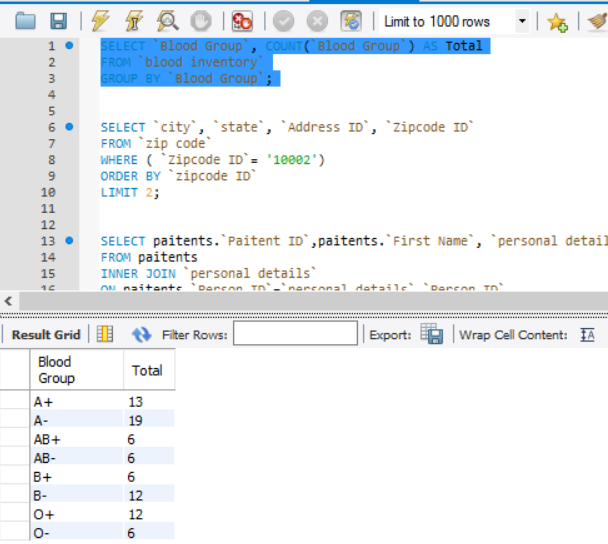
**Queries:**

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 Bank ID`;



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

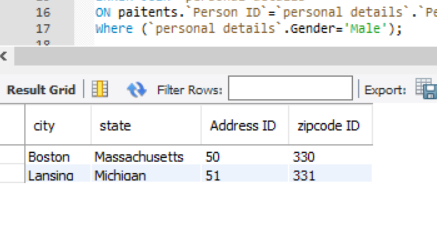
SELECT `city`, `state`, `Address ID`,`zipcode ID`

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,

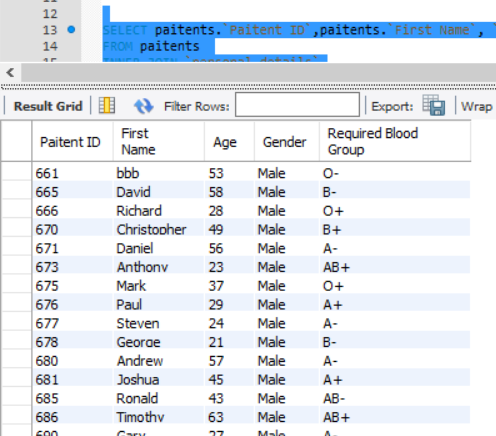
`personal details`.`Blood Group` AS `Required Blood Group`

FROM paitents

INNER JOIN `personal details`

ON paitents.`Person ID`=`personal details`.`Person ID`

WHERE (`personal details`.Gender='Male');



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

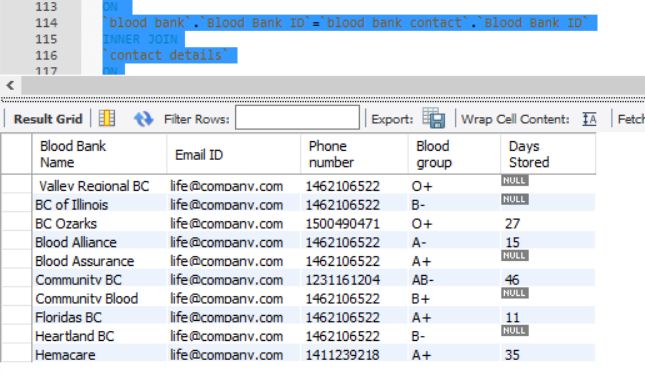
`contact details`

ON

`blood bank contact`.`Contact ID`=`contact details`.`Contact ID`

GROUP BY `blood bank`.`Blood Bank Name`

LIMIT 10;



5)Transaction and RollBack

START TRANSACTION;

UPDATE `blood inventory` SET `Blood group` = 'B+'

WHERE `Blood ID` = 121;

UPDATE `blood inventory` SET price = '1000'

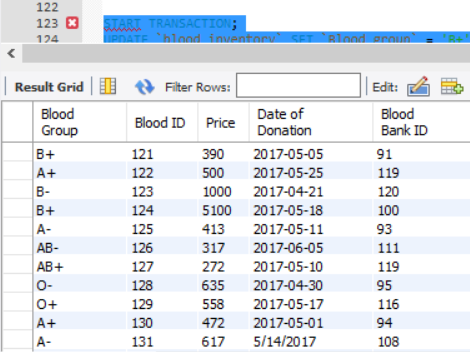
WHERE `Blood ID` = 123;

SELECT \* FROM `blood inventory`;

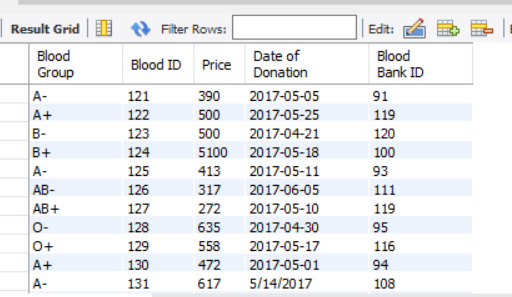
ROLLBACK;

SELECT \* FROM `blood inventory`;

Transaction:



After RollBack :



**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);

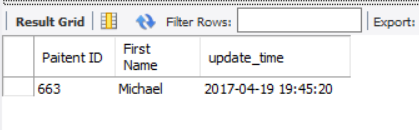
end\\

-----test-----

update paitents set `First Name`='sam'

where `Paitent ID`=663;

select \* from UpdatePaitentDetail;



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'

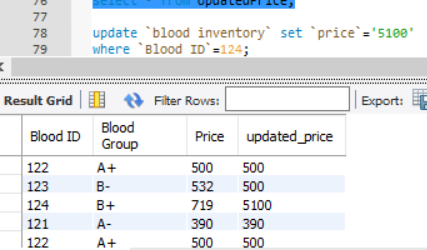
where `Blood ID`=122;

update `blood inventory` set `price`='500'

where `Blood ID`=123;

update `blood inventory` set `price`='5100'

where `Blood ID`=124;



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

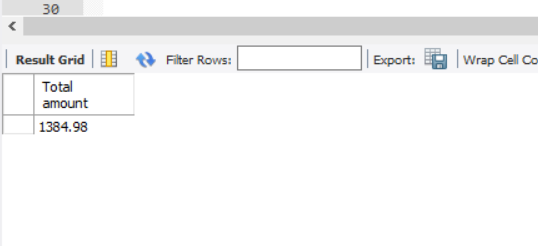
SET @sum = new.price + new.tax;

SET @sum=0;

INSERT INTO Total\_Bill(price, Tax)

VALUES(1370,14.98);

SELECT @sum AS 'Total amount';



**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

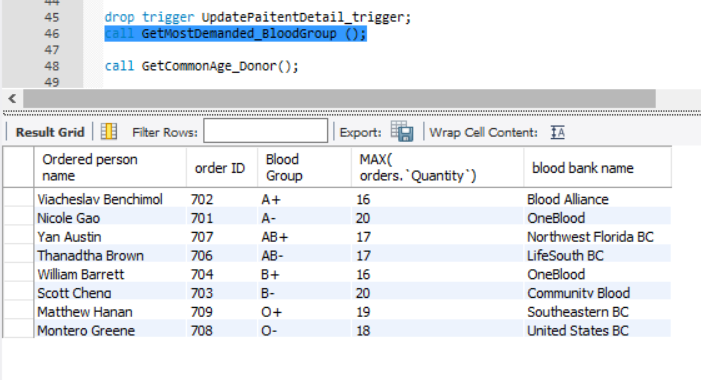
orders.`hospital employee ID`=admistration.`hospital employee ID`

GROUP BY orders.`Blood Group`;

END

-----test-----

-🡪 call MaxQuantityOrder \_BloodGroup ();



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`

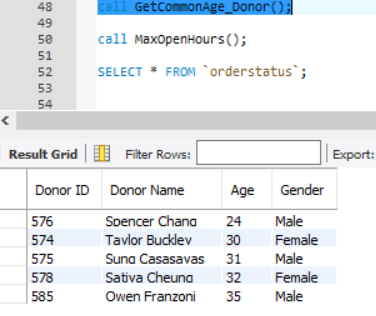
WHERE `personal details`.`Age` between 22 and 35

GROUP BY `personal details`.`Age`;

END

----test-----

Call GetCommonAge\_Donor();



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`

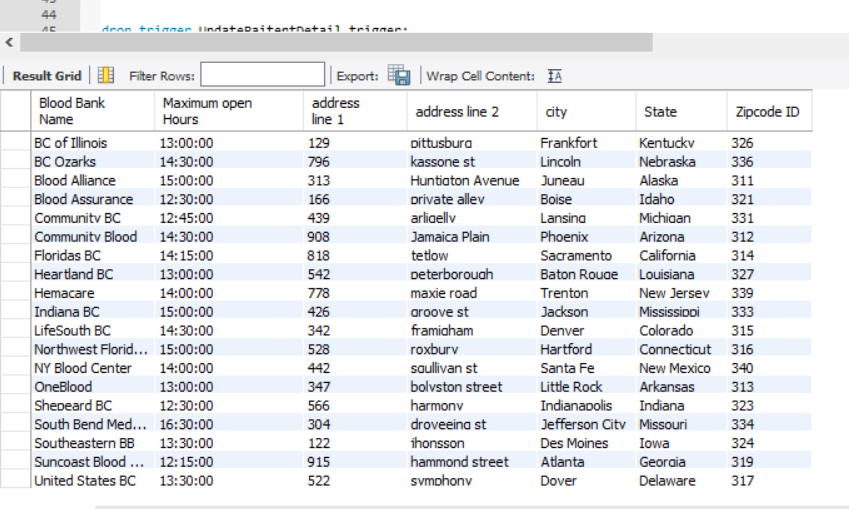
WHERE Timediff(Timings.`close timings`,Timings.`Open timings`)> '12:00:00'

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

END

----test-----

Call MaxOpenHours()



**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`

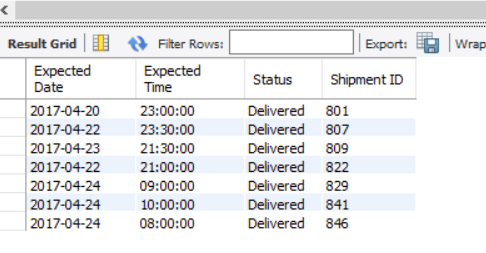
FROM

`status`

WHERE

((`status`.`Status` = 'Delivered')

AND (CAST(`status`.`Expected Date` AS DATE) < '2017-04-25')));



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

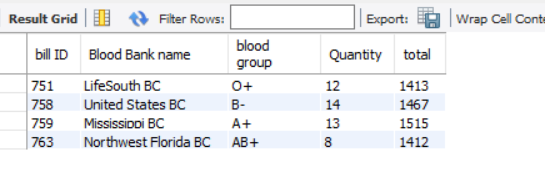
(`orders`

JOIN `billing information` ON ((`orders`.`Order ID` = `billing information`.`Order ID`)))

WHERE

(`billing information`.`Total` > 1400)

LIMIT 4)



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

;