



# **PROJECT REPORT**

A report submitted to the

Department of Electrical and Information Engineering  
Faculty of Engineering  
University of Ruhuna

Sri Lanka

On 09<sup>th</sup> of April 2024

In completing a Report for the module  
EE4350 Database Systems

By:

|                      |                |
|----------------------|----------------|
| JACKSHAN VENUJAN G.S | - EG/2021/4566 |
| JATHURSHAN T.        | - EG/2021/4568 |
| JAYAKODY J.A.C.I     | - EG/2021/4569 |

# Table of Contents

|  |    |
|--|----|
| Chapter 1: Requirement Analysis .....                            | 3  |
| 1.1.    Functional Requirements .....                            | 3  |
| 1.2.    Data Requirements .....                                  | 3  |
| Chapter 2: Conceptual Design .....                               | 6  |
| 2.1.    ER Diagram.....  | 6  |
| 2.2.    ER Diagram using Visual Paradigm tool .....              | 6  |
| 2.3.    EER Diagram Using Workbench .....                        | 7  |
| 2.4.    UML Class Diagram .....                                  | 7  |
| Chapter 3: Implementation.....                                   | 8  |
| 3.1.    Schema Creation .....                                    | 8  |
| 3.2.    Creation of Tables & Insertion of data.....              | 9  |
| 3.3.    Table Alteration & Deriving Age from Date of Birth ..... | 23 |
| 3.4.    Table Definitions.....                                   | 24 |
| 3.5.    Updating & Deleting data of tables.....                  | 39 |
| Chapter 4: Transactions.....                                     | 54 |
| 4.1.    Simple Queries .....                                     | 54 |
| 4.2.    Complex Queries.....                                     | 58 |
| Chapter 5: Tuning.....   | 65 |
| 5.1.    Tuning of Union Operation .....                          | 66 |
| 5.2.    Tuning of Intersection Operation .....                   | 67 |
| 5.3.    Tuning of Set Difference Operation.....                  | 68 |
| 5.4.    Tuning of Natural Join Operation .....                   | 69 |
| 5.5.    Tuning of Right Outer Join Operation .....               | 70 |
| 5.6.    Tuning of Left Outer Join Operation .....                | 71 |
| 5.7.    Tuning of Full Outer Union Operation .....               | 72 |
| 5.8.    Tuning of Nested Query 1 .....                           | 73 |
| 5.9.    Tuning of Nested Query 2 .....                           | 74 |
| 5.10.    Tuning of Nested Query 3 .....                          | 75 |

# Chapter 1: Requirement Analysis

## 1.1. Functional Requirements

- Functional requirement analysis for a blood bank management system is important for ensuring efficient operations in blood donation and transfusion processes. This database system serves as a centralized platform for managing donor information, blood inventory, and recipient details. Its primary objective is to organize the donation process, facilitate timely allocation of blood units to recipients, and maintain accurate records for tracking donations and transfusions. By automating these tasks like donor registration, blood inventory tracking, scheduling donation camps, and providing quick solutions to patients, this database system aims to improve the overall effectiveness and accessibility of blood banking services from all over its branches in the island.
- In this system, we required 9 entities, including 7 strong entities such as “BLOOD”, “BLOOD BANK”, “HOSPITAL”, “EMPLOYEE”, “DONAR”, “PAYMENT”, “PATIENT” and 2 weak entities such as “EQUIPMENT” and “BLOOD DONATION CAMPS”. All these entities were interconnected with appropriate relationships according to the requirements. The Entity-Relationship Diagram which explains the relationship between the entities is illustrated in chapter 2.

- 01) Registration Process: Add new donors, employees, patients and hospitals to the records.
- 02) Updating the entries: Modify donors, employees, patients and hospitals records in the system.
- 03) Deleting the entries: Delete the records form the system
- 04) Data retravel: Using queries get the information needed from the system

## 1.2. Data Requirements

Attributes of Entity “BLOOD”

|                               |                      |
|-------------------------------|----------------------|
| • Blood_ID                    | (Should not be NULL) |
| • Blood_Group                 | (Should not be NULL) |
| • Blood_Quantity              |                      |
| • Expiry_Date                 | (Should not be NULL) |
| • Blood_Description           |                      |
| • Test_Done_for_Blood_Samples |                      |

Attributes of Entity “BLOOD BANK”

|                            |                      |
|----------------------------|----------------------|
| • Branch_ID                | (Should not be NULL) |
| • Branch_Name              | (Should not be NULL) |
| • Available_Blood_Groups   |                      |
| • Available_Blood_Quantity |                      |
| • Street                   | (Should not be NULL) |
| • City                     | (Should not be NULL) |
| • Province                 | (Should not be NULL) |
| • Contact_No               | (Should not be NULL) |

Attributes of Entity “HOSPITAL”

|                            |                      |
|----------------------------|----------------------|
| • Hospital_ID              | (Should not be NULL) |
| • Hospital_Name            | (Should not be NULL) |
| • Available_Blood_Groups   |                      |
| • Available_Blood_Quantity |                      |
| • Street                   | (Should not be NULL) |
| • City                     | (Should not be NULL) |
| • Province                 | (Should not be NULL) |
| • Contact_No               | (Should not be NULL) |

Attributes of Entity “EMPLOYEE”

|                 |                      |
|-----------------|----------------------|
| • Employee_ID   | (Should not be NULL) |
| • Employee_Name | (Should not be NULL) |
| • Employee_Type | (Should not be NULL) |
| • Gender        | (Should not be NULL) |
| • Date_of_Birth | (Should not be NULL) |
| • Nationality   | (Should not be NULL) |
| • Salary        | (Should not be NULL) |
| • Age           | (Should not be NULL) |
| • Street        | (Should not be NULL) |
| • City          | (Should not be NULL) |
| • Province      | (Should not be NULL) |
| • Contact_No    | (Should not be NULL) |
| • Email         |                      |

Attributes of Entity “EQUIPMENT”

|                     |                      |
|---------------------|----------------------|
| • Equip_Name        | (Should not be NULL) |
| • Cost              | (Should not be NULL) |
| • Quantity          |                      |
| • Equip_Description | (Should not be NULL) |

Attributes of Entity “BLOOD DONATION CAMPS”

|             |                      |
|-------------|----------------------|
| • Camp_Name | (Should not be NULL) |
| • Date      | (Should not be NULL) |
| • Street    | (Should not be NULL) |
| • City      | (Should not be NULL) |
| • Province  | (Should not be NULL) |

Attributes of Entity “DONOR”

|                          |                      |
|--------------------------|----------------------|
| • Donor_ID               | (Should not be NULL) |
| • Donor_Name             | (Should not be NULL) |
| • Gender                 |                      |
| • Blood_Group            | (Should not be NULL) |
| • Date_of_Birth          |                      |
| • Age                    |                      |
| • Donated_Blood_Quantity |                      |
| • Donated_Dates          |                      |
| • Eligibility_Status     | (Should not be NULL) |
| • Last_Donated_Date      |                      |
| • Nationality            | (Should not be NULL) |
| • Health_Problems        |                      |
| • Street                 | (Should not be NULL) |
| • City                   | (Should not be NULL) |
| • Province               | (Should not be NULL) |
| • Contact_No             | (Should not be NULL) |
| • Email                  |                      |

Attributes of Entity “PAYMENT”

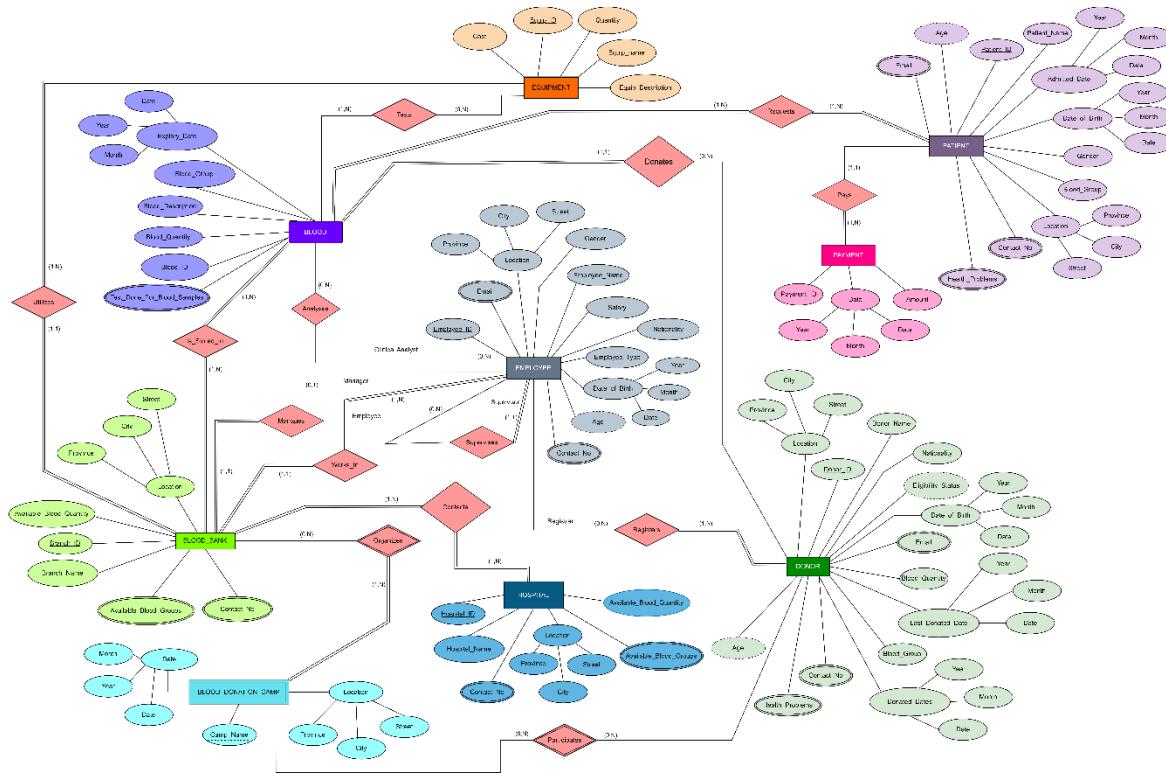
|              |                      |
|--------------|----------------------|
| • Payment_ID | (Should not be NULL) |
| • Amount     | (Should not be NULL) |
| • Paid_Date  | (Should not be NULL) |

Attributes of Entity “PATIENT”

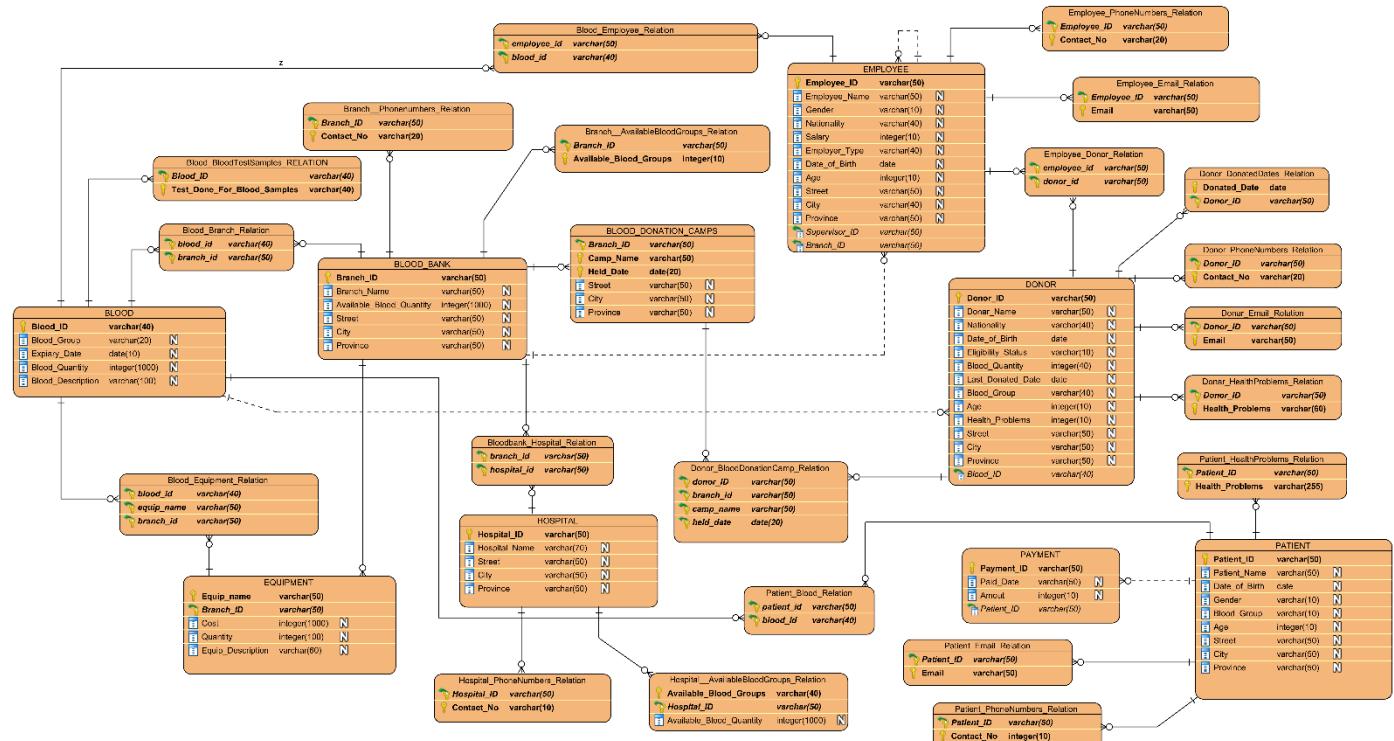
|                   |                      |
|-------------------|----------------------|
| • Patient_ID      | (Should not be NULL) |
| • Patient_Name    | (Should not be NULL) |
| • Gender          | (Should not be NULL) |
| • Blood_Group     | (Should not be NULL) |
| • Date_of_Birth   | (Should not be NULL) |
| • Age             | (Should not be NULL) |
| • Nationality     | (Should not be NULL) |
| • Health_Problems |                      |
| • Street          | (Should not be NULL) |
| • City            | (Should not be NULL) |
| • Province        | (Should not be NULL) |
| • Contact_No      | (Should not be NULL) |
| • Email           |                      |
| • Admitted_Date   | (Should not be NULL) |

## **Chapter 2: Conceptual Design**

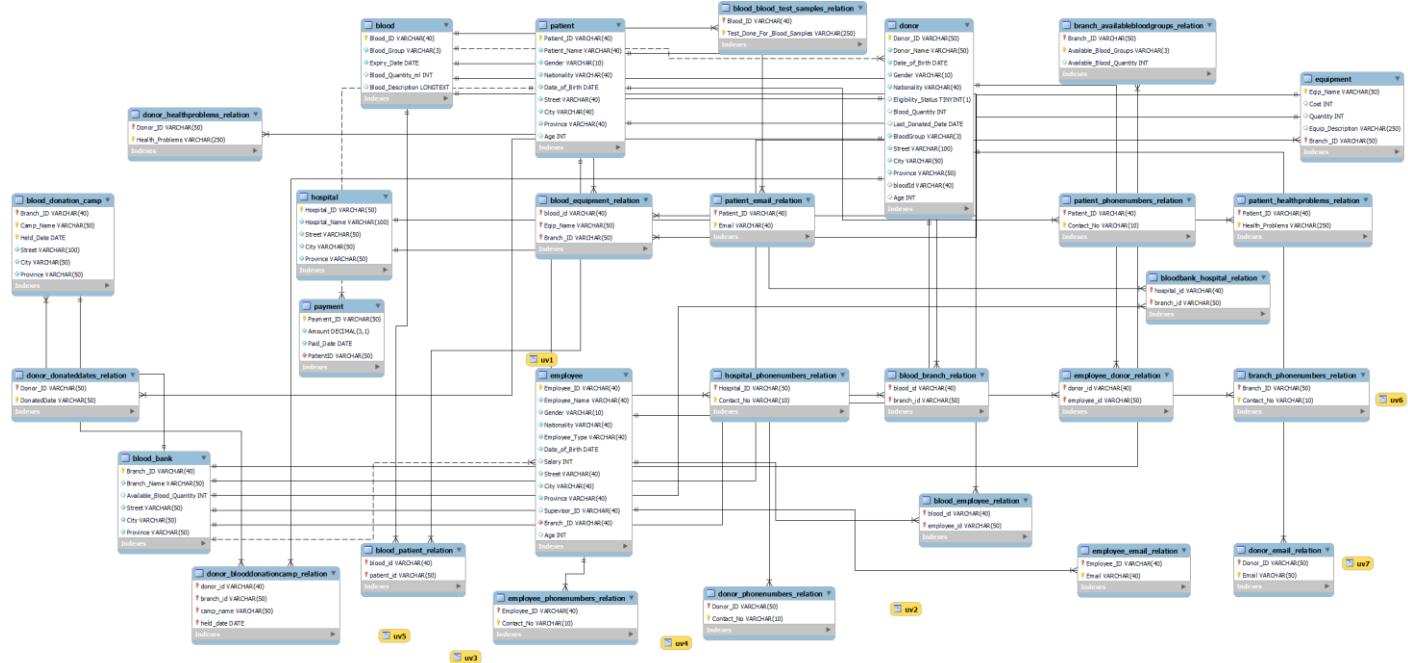
## 2.1. ER Diagram



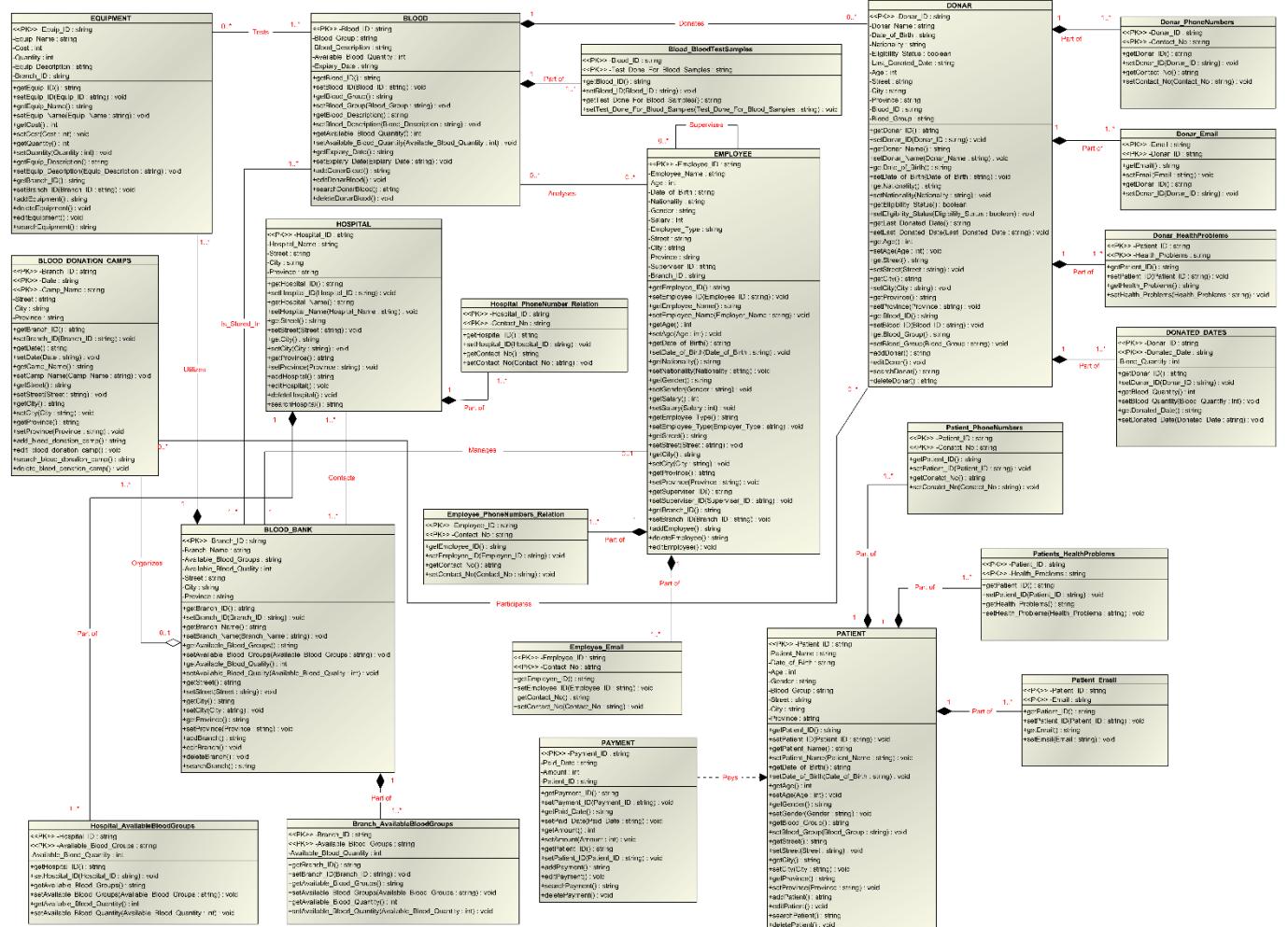
## 2.2. ER Diagram using Visual Paradigm tool



### 2.3. EER Diagram Using Workbench



## 2.4. UML Class Diagram



# Chapter 3: Implementation

## 3.1. Schema Creation

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the **SCHEMAS** section with databases: `blood_bank_management_system`, `foe_23`, `sakila`, `sys`, and `world`.
- Query Editor:** Displays the following SQL script:

```
1 • CREATE DATABASE blood_bank_management_system;
2 • SHOW DATABASES;
3 • USE blood_bank_management_system;
```
- Result Grid:** Shows the list of databases in the current schema:

| Database                                  |
|---|
| <code>blood_bank_management_system</code> |
| <code>foe_23</code>                       |
| <code>information_schema</code>           |
| <code>mysql</code>                        |
| <code>performance_schema</code>           |
| <code>sakila</code>                       |
| <code>sys</code>                          |
| <code>world</code>                        |
- Output:** Shows the execution log for the queries:

| # | Time     | Action   | Message   | Duration / Fetch |
|---|----------|--|---|------------------|
| 1 | 14:52:57 | CREATE DATABASE blood_bank_management_system   | Error Code: 1007: Can't create database 'blood_bank_management_system'; database exists | 0.000 sec        |
| 2 | 14:53:04 | USE blood_bank_management_system   | 0 row(s) affected   | 0.000 sec        |
| 3 | 14:53:05 | CREATE TABLE blood_bank (id INT AUTO_INCREMENT NOT NULL, donor_id VARCHAR(50) NOT NULL, contact_no VARCHAR(10) NOT NULL, PRIMARY KEY (id)) ENGINE=InnoDB | 0 row(s) affected   | 0.079 sec        |

## 3.2. Creation of Tables & Insertion of data

### 3.1.1. Blood Table

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** blood\_bank\_management\_system
- Table:** BLOOD
- Query Editor:** Contains the SQL code for creating the BLOOD table and inserting 20 rows of blood bank data.
- Result Grid:** Displays the inserted data in a grid format.
- Data:** The result grid shows 20 rows of blood bank data with columns: Blood\_ID, Blood\_Group, Expiry\_Date, Blood\_Quantity\_ml, and Blood\_Description.

| Blood_ID | Blood_Group | Expiry_Date | Blood_Quantity_ml | Blood_Description |
|----------|-------------|-------------|-------------------|-------------------|
| B33467   | O+          | 2024-06-25  | 390               | stored            |
| B33469   | A+          | 2024-07-11  | 500               | stored            |
| B33470   | B+          | 2024-06-15  | 450               | stored            |
| B33486   | A-          | 2024-05-20  | 400               | stored            |
| B33492   | O-          | 2024-06-02  | 385               | stored            |
| B33493   | O+          | 2024-06-25  | 470               | stored            |
| B33500   | O-          | 2024-08-07  | 430               | stored            |
| B33467   | O+          | 2024-06-25  | 390               | stored            |
| B33469   | A+          | 2024-07-11  | 500               | stored            |
| B33470   | B+          | 2024-06-15  | 450               | stored            |
| B33486   | A-          | 2024-05-20  | 400               | stored            |
| B33492   | O-          | 2024-06-02  | 385               | stored            |
| B33493   | O+          | 2024-06-25  | 470               | stored            |
| B33500   | O-          | 2024-08-07  | 430               | stored            |
| B33467   | O+          | 2024-06-25  | 390               | stored            |
| B33469   | A+          | 2024-07-11  | 500               | stored            |
| B33470   | B+          | 2024-06-15  | 450               | stored            |
| B33486   | A-          | 2024-05-20  | 400               | stored            |
| B33492   | O-          | 2024-06-02  | 385               | stored            |
| B33493   | O+          | 2024-06-25  | 470               | stored            |
| B33500   | O-          | 2024-08-07  | 430               | stored            |

### 3.1.3. Blood - Branch Relation Table

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** blood\_branch\_relation
- Table:** Blood\_Branch\_Relation
- Query Editor:** Contains the SQL code for creating the Blood\_Branch\_Relation table and inserting 24 rows of blood branch relation data.
- Result Grid:** Displays the inserted data in a grid format.
- Data:** The result grid shows 24 rows of blood branch relation data with columns: blood\_id and branch\_id.

| blood_id | branch_id |
|----------|-----------|
| B33467   | B003      |
| B33493   | B003      |
| B33470   | B009      |
| B33467   | B011      |
| B33469   | B014      |
| B33492   | B016      |
| B33486   | B021      |
| B33469   | B024      |
| B33467   | B024      |
| B33467   | B003      |
| B33493   | B003      |
| B33470   | B009      |
| B33467   | B011      |
| B33469   | B014      |
| B33492   | B016      |
| B33486   | B021      |
| B33469   | B024      |
| B33467   | B024      |

#### **3.1.4. Blood - Blood Donation Camps Table**

MySQL Workbench

Local instance MySQL80 (blo...)

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- blood\_patient\_relation
- bloodbank\_hospital\_relation
- branch\_availablebloodgroup
- branch\_phonenumbers\_relation
- donor
- donor\_blooddonationcamp\_relation
- donor\_donateddates\_relation
- donor\_email\_relation
- donor\_healthproblems\_relation
- donor\_phonenumbers\_relation
- employee
- employee\_donor\_relation
- employee\_email\_relation
- employee\_phonenumbers\_relation
- equipment
- hospital
- hospital\_phonenumbers\_relation
- patient

Administration Schemas

Information

Table: donor\_blooddonationcamp\_relation

Columns:

| donor_id | branch_id | camp_name | held_date  |
|----------|-----------|-----------|------------|
| D651355  | Br003     | Camp2     | 2024-02-10 |
| D651354  | Br011     | Camp1     | 2024-01-30 |
| D651360  | Br014     | Camp4     | 2024-01-22 |
| D651369  | Br016     | Camp8     | 2024-01-02 |
| D651362  | Br021     | Camp      | 2024-03-12 |
| D651357  | Br024     | Camp3     | 2024-01-13 |
| D651366  | Br037     | Camp7     | 2024-02-20 |
|          |           |           |            |

Blood Bank Management System.

CREATE TABLE BLOOD\_DONATION\_CAMP(

Branch\_ID varchar(48) not null,

Camp\_Name varchar(50) not null,

Held\_Date date not null ,

Street varchar(100) not null,

City varchar(50) not null,

Province varchar(50) not null,

PRIMARY KEY(Branch\_ID,Camp\_Name,Held\_Date),

constraint fk\_branch foreign key(Branch\_ID) references Blood\_Bank(Branch\_ID) ON DELETE CASCADE

)

insert into BLOOD\_DONATION\_CAMP values('Br011','Camp1','2024-01-30','Street1','Anuradhapura','North Central');

insert into BLOOD\_DONATION\_CAMP values('Br003','Camp2','2024-02-10','Street2','Ampara','Eastern');

insert into BLOOD\_DONATION\_CAMP values('Br024','Camp3','2024-01-13','Street3','Jaffna','Northern');

insert into BLOOD\_DONATION\_CAMP values('Br014','Camp4','2024-01-22','Street4','Kurunegala','North Western');

insert into BLOOD\_DONATION\_CAMP values('Br009','Camp5','2024-02-17','Streets','Badulla','Uva');

insert into BLOOD\_DONATION\_CAMP values('Br021','Camp6','2024-03-12','Street6','Peradeniya','Central');

insert into BLOOD\_DONATION\_CAMP values('Br037','Camp7','2024-02-20','Street7','Kandy','Central');

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

| donor_id | branch_id | camp_name | held_date  |
|----------|-----------|-----------|------------|
| D651355  | Br003     | Camp2     | 2024-02-10 |
| D651354  | Br011     | Camp1     | 2024-01-30 |
| D651360  | Br014     | Camp4     | 2024-01-22 |
| D651369  | Br016     | Camp8     | 2024-01-02 |
| D651362  | Br021     | Camp      | 2024-03-12 |
| D651357  | Br024     | Camp3     | 2024-01-13 |
| D651366  | Br037     | Camp7     | 2024-02-20 |
|          |           |           |            |

donor\_blooddonationcamp\_relation

Object Info Session

Query Completed

Output

Result Grid

Form Editor

Apply Revert

### **3.1.5. Blood- Employee Relation Table**

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the Schemas tree with various tables like blood\_branch\_relation, blood\_donation\_camp, etc.
- Current Schema:** Blood Bank Management System..
- Table:** blood\_employee\_relation
- Code Editor:** The code pane displays the SQL script for creating the table and inserting data. The table structure is as follows:

| blood_id | employee_id |
|----------|-------------|
| B33467   | E034        |
| B33469   | E045        |
| B33466   | E045        |
| B33470   | E053        |
| B33500   | E058        |
| B33493   | E066        |
| B33467   | E072        |
| B33492   | E072        |
| B33491   | E072        |

- Result Grid:** The result grid shows the inserted data:

| blood_id | employee_id |
|----------|-------------|
| B33467   | E034        |
| B33469   | E045        |
| B33466   | E045        |
| B33470   | E053        |
| B33500   | E058        |
| B33493   | E066        |
| B33467   | E072        |
| B33492   | E072        |
| B33491   | E072        |

- Status:** DONE
- Bottom Navigation:** Object Info, Session, Output, Apply, Revert.

### 3.1.6. Blood- Equipment Relation Table

MySQL Workbench

Local instance MySQL80 (blo...)

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

Table: blood\_equipment\_relation

Columns:

- blood\_id** varchar(40) PK
- Eqip\_Name** varchar(50) PK
- Branch\_ID** varchar(50) PK

```

416 • CREATE TABLE Blood_Equipment_Relation(
417     blood_id varchar(40) not null,
418     Eqip_Name varchar(50) not null,
419     Branch_ID varchar(50) not null,
420     PRIMARY KEY(Eqip_Name,Branch_ID,blood_id),
421     constraint fk_blood2 foreign key(blood_id) references BLOOD(Blood_ID)ON DELETE CASCADE ON UPDATE CASCADE,
422     constraint fk_equipment foreign key(eqip_name) references EQUIPMENT(Eqip_Name) ON DELETE CASCADE ON UPDATE CASCADE,
423     constraint fk_branch8 foreign key(branch_id) references EQUIPMENT(Branch_ID) ON DELETE CASCADE ON UPDATE CASCADE
424   );
425 •   SELECT * FROM blood_bank_management_system.blood_equipment_relation;
426 •   insert into Blood_Equipment_Relation values('B33467','Blood collection monitor','Br011');
427 •   insert into Blood_Equipment_Relation values('B33467','Blood bank refrigerator','Br003');
428 •   insert into Blood_Equipment_Relation values('B33469','Blood donor couch','Br024');
429 •   insert into Blood_Equipment_Relation values('B33469','Blood bag tube sealer','Br014');
430 •   insert into Blood_Equipment_Relation values('B33469','Blood mixer roller','Br009');
431 •   insert into Blood_Equipment_Relation values('B33470','Blood cell separator machine','Br011');
432 •   insert into Blood_Equipment_Relation values('B33470','Blood filters','Br021');
433 •   insert into Blood_Equipment_Relation values('B33486','Blood circulation machine','Br016');

```

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

| blood_id | Eqip_Name                    | Branch_ID |
|----------|------------------------------|-----------|
| B33467   | Blood bank refrigerator      | Br003     |
| B33467   | Blood collection monitor     | Br011     |
| B33469   | Blood bag tube sealer        | Br014     |
| B33469   | Blood donor couch            | Br024     |
| B33469   | Blood mixer roller           | Br009     |
| B33470   | Blood cell separator machine | Br011     |
| B33470   | Blood filters                | Br021     |
| B33486   | Blood circulation machine    | Br016     |

blood\_equipment\_relation 23 x

Object Info Session Output

Query Completed

### 3.1.7. Blood- Patient Table

MySQL Workbench

Local instance MySQL80 (blo...)

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

Table: blood\_patient\_relation

Columns:

- blood\_id** varchar(40) PK
- patient\_id** varchar(50) PK

```

530 • CREATE TABLE Blood_Patient_Relation(
531     blood_id varchar(40) not null,
532     patient_id varchar(50) not null,
533     PRIMARY KEY(blood_id,patient_id),
534     constraint fk_blood5 foreign key(blood_id) references BLOOD(Blood_ID)ON DELETE CASCADE ON UPDATE CASCADE,
535     constraint fk_patient4 foreign key(patient_id) references PATIENT(Patient_ID) ON DELETE CASCADE ON UPDATE CASCADE
536   );
537 •   insert into Blood_Patient_Relation values('B33467','P23487');
538 •   insert into Blood_Patient_Relation values('B33470','P23488');
539 •   insert into Blood_Patient_Relation values('B33486','P23489');
540 •   insert into Blood_Patient_Relation values('B33493','P23490');
541 •   insert into Blood_Patient_Relation values('B33500','P23491');
542 •   insert into Blood_Patient_Relation values('B33500','P23491');

```

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

| blood_id | patient_id |
|----------|------------|
| B33467   | P23487     |
| B33470   | P23488     |
| B33486   | P23489     |
| B33493   | P23490     |
| B33500   | P23491     |

blood\_patient\_relation 29 x

Object Info Session Output

Query Completed

### 3.1.8. Blood- Test Samples Relation Table

MySQL Workbench

Local instance MySQL80 (blo...)

**Schemas:**

- blood\_bank\_management\_system
  - Tables
    - blood
    - blood\_bank
    - blood\_bloodtestsamples\_relation
    - blood\_branch\_relation
    - blood\_donation\_center
    - blood\_employee\_relation
    - blood\_equipment\_relation
    - blood\_patient\_relation
    - bloodbank\_hospital\_relation
    - branch\_availablebloodgroup
    - branch\_phonenumbers\_relation
    - donor
    - donor\_blooddonationcenter
    - donor\_donateddates\_relation
    - donor\_email\_relation
    - donor\_healthproblems\_relation

**Table: blood\_bloodtestsamples\_relation**

**Columns:**

- Blood\_ID
- Test\_Done\_For\_Blood\_Samples

```

CREATE TABLE Blood_BloodTestSamples_Relation (
    Blood_ID varchar(40),
    Test_Done_For_Blood_Samples varchar(250),
    PRIMARY KEY (Blood_ID, Test_Done_For_Blood_Samples),
    CONSTRAINT fk_blood FOREIGN KEY (Blood_ID) REFERENCES BLOOD(Blood_ID) ON DELETE CASCADE ON UPDATE CASCADE
);

insert into Blood_BloodTestSamples_Relation values('B33467','Lipid Panel');
insert into Blood_BloodTestSamples_Relation values('B33469','Complete Blood Count');
insert into Blood_BloodTestSamples_Relation values('B33469','Basic Metabolic Panel');
insert into Blood_BloodTestSamples_Relation values('B33470','Basic Metabolic Panel');
insert into Blood_BloodTestSamples_Relation values('B33486','Lipid Panel');
insert into Blood_BloodTestSamples_Relation values('B33486','Complete Blood Count');
insert into Blood_BloodTestSamples_Relation values('B33493','Complete Blood Count');
insert into Blood_BloodTestSamples_Relation values('B33493','Arterial Blood Gas');
insert into Blood_BloodTestSamples_Relation values('B33493','Basic Metabolic Panel');
insert into Blood_BloodTestSamples_Relation values('B33493','Complete Blood Count');

```

**Result Grid:**

| Blood_ID | Test_Done_For_Blood_Samples |
|----------|-----------------------------|
| B33467   | Lipid Panel                 |
| B33469   | Basic Metabolic Panel       |
| B33469   | Complete Blood Count        |
| B33470   | Basic Metabolic Panel       |
| B33486   | Complete Blood Count        |
| B33486   | Lipid Panel                 |
| B33493   | Arterial Blood Gas          |
| B33493   | Basic Metabolic Panel       |
| B33493   | Complete Blood Count        |

### 3.1.9. Blood Bank Table

MySQL Workbench

Local instance MySQL80 (blo...)

**Schemas:**

- blood\_bank\_management\_system
  - Tables
    - blood
    - blood\_bank
    - blood\_bloodtestsamples\_relation
    - blood\_branch\_relation
    - blood\_donation\_center
    - blood\_employee\_relation
    - blood\_equipment\_relation
    - blood\_patient\_relation
    - bloodbank\_hospital\_relation
    - branch\_availablebloodgroup
    - branch\_phonenumbers\_relation
    - donor
    - donor\_blooddonationcenter
    - donor\_donateddates\_relation
    - donor\_email\_relation
    - donor\_healthproblems\_relation

**Table: blood\_bloodtestsamples\_relation**

**Columns:**

- Blood\_ID
- Test\_Done\_For\_Blood\_Samples

```

CREATE TABLE BLOOD_BANK(
    Branch_ID varchar(40) not null,
    Branch_Name varchar(50) not null,
    Available_Blood_Quantity int default 0,
    Street varchar(50) not null,
    City varchar(50) not null,
    Province varchar(50) not null,
    PRIMARY KEY(Branch_ID)
);

SELECT * FROM blood_bank_management_system.blood_bank;
insert into BLOOD_BANK values('Br011','Anuradhapura',7500,'Street1','Anuradhapura','North Central');
insert into BLOOD_BANK values('Br003','Mahoya',8960,'Street2','Ampara','Eastern');
insert into BLOOD_BANK values('Br024','Kilinochchi',3400,'Street3','Jaffna','Northern');
insert into BLOOD_BANK values('Br014','Dambadeniya',4830,'Street4','Kurunegala','North Western');
insert into BLOOD_BANK values('Br009','Diyathalawa',5400,'Street5','Badulla','Uva');
insert into BLOOD_BANK values('Br021','Gampola',6890,'Street6','Peradeniya','Central');
insert into BLOOD_BANK values('Br037','Dambulla',0,'Street7','Kandy','Central');
insert into BLOOD_BANK values('Br016','Army Hospital',9850,'Street8','Colombo','Western');

```

**Result Grid:**

| Branch_ID | Branch_Name   | Available_Blood_Quantity | Street  | City         | Province      |
|-----------|---------------|--------------------------|---------|--------------|---------------|
| Br003     | Mahoya        | 8960                     | Street2 | Ampara       | Eastern       |
| Br009     | Diyathalawa   | 5400                     | Street5 | Badulla      | Uva           |
| Br011     | Anuradhapura  | 7500                     | Street1 | Anuradhapura | North Central |
| Br014     | Dambadeniya   | 4830                     | Street4 | Kurunegala   | North Western |
| Br016     | Army Hospital | 9850                     | Street  | Colombo      | Western       |
| Br021     | Gampola       | 6890                     | Street  | Peradeniya   | Central       |
| Br024     | Kilinochchi   | 3400                     | Street3 | Jaffna       | Northern      |
| Br037     | Dambulla      | 0                        | Street7 | Kandy        | Central       |
| NULL      | NULL          | NULL                     | NULL    | NULL         | NULL          |

### 3.1.10. Blood Bank- Hospital Relation Table

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** blood\_bank\_management\_system
- Table:** blood\_branch\_relation
- Columns:**
  - blood\_id: varchar(40) PK
  - branch\_id: varchar(50) PK
- Code:** SQL script for creating the table and inserting data into the Bloodbank\_Hospital\_Relation table.
- Result Grid:** Shows the inserted data with columns hospital\_id and branch\_id.

```

CREATE TABLE Bloodbank_Hospital_Relation(
    hospital_id varchar(40) not null,
    branch_id varchar(50) not null,
    PRIMARY KEY(hospital_id,branch_id),
    constraint fk_hospital2 foreign key(hospital_id) references HOSPITAL(Hospital_ID) ON DELETE CASCADE ON UPDATE CASCADE,
    constraint fk_branch7 foreign key(branch_id) references BLOOD_BANK(Branch_ID) ON DELETE CASCADE ON UPDATE CASCADE
);

insert into Bloodbank_Hospital_Relation values('H008','Br011');
insert into Bloodbank_Hospital_Relation values('H008','Br003');
insert into Bloodbank_Hospital_Relation values('H010','Br024');
insert into Bloodbank_Hospital_Relation values('H011','Br014');
insert into Bloodbank_Hospital_Relation values('H011','Br009');
insert into Bloodbank_Hospital_Relation values('H015','Br021');
insert into Bloodbank_Hospital_Relation values('H016','Br037');
insert into Bloodbank_Hospital_Relation values('H019','Br018');
insert into Bloodbank_Hospital_Relation values('H019','Br011');
insert into Bloodbank_Hospital_Relation values('H022','Br009');

```

| hospital_id | branch_id |
|-------------|-----------|
| H008        | Br003     |
| H011        | Br009     |
| H022        | Br009     |
| H008        | Br011     |
| H019        | Br011     |
| H011        | Br014     |
| H019        | Br016     |
| H015        | Br021     |
| H019        | Br011     |

### 3.1.11. Branch- Available Blood Groups Table

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** blood\_bank\_management\_system
- Table:** branch\_availablebloodgroups\_relation
- Columns:**
  - Branch\_ID: varchar(50) PK
  - Available\_Blood\_Groups: varchar(3) PK
  - Available\_Blood\_Quantity: int
- Code:** SQL script for creating the table and inserting data into the Branch\_AvailableBloodGroups\_Relation table.
- Result Grid:** Shows the inserted data with columns Branch\_ID, Available\_Blood\_Groups, and Available\_Blood\_Quantity.

```

CREATE TABLE Branch_AvailableBloodGroups_Relation(
    Branch_ID varchar(50) not null,
    Available_Blood_Groups varchar(3) not null,
    Available_Blood_Quantity int default 0,
    primary key(Branch_ID,Available_Blood_Groups),
    constraint fk_bloodbank foreign key(Branch_ID) references BLOOD_BANK(Branch_ID) ON DELETE CASCADE ON UPDATE CASCADE
);

insert into Branch_AvailableBloodGroups_Relation values('Br011','O-',5000);
insert into Branch_AvailableBloodGroups_Relation values('Br011','B-',2500);
insert into Branch_AvailableBloodGroups_Relation values('Br003','O+',8960);
insert into Branch_AvailableBloodGroups_Relation values('Br024','A+',3400);
insert into Branch_AvailableBloodGroups_Relation values('Br014','O+',2200);
insert into Branch_AvailableBloodGroups_Relation values('Br014','O-',1500);
insert into Branch_AvailableBloodGroups_Relation values('Br014','A-',1130);
insert into Branch_AvailableBloodGroups_Relation values('Br009','O-',5400);
insert into Branch_AvailableBloodGroups_Relation values('Br021','O+',4000);

```

| Branch_ID | Available_Blood_Groups | Available_Blood_Quantity |
|-----------|------------------------|--------------------------|
| Br003     | O+                     | 8960                     |
| Br009     | O-                     | 5400                     |
| Br011     | B-                     | 2500                     |
| Br011     | O-                     | 5000                     |
| Br014     | A-                     | 1130                     |
| Br014     | O-                     | 1500                     |
| Br014     | O+                     | 2200                     |
| Br016     | A-                     | 4000                     |
| Br014     | A-                     | 1750                     |

### 3.1.12. Branch- Phone Numbers Table

```

CREATE TABLE Branch_Phonenumbers_Relation(
    Branch_ID varchar(50) not null,
    Contact_No varchar(10) not null,
    primary key(Branch_ID,Contact_No),
    constraint fk_bloodbank1 foreign key(Branch_ID) references BLOOD_BANK(Branch_ID)
    ON DELETE CASCADE ON UPDATE CASCADE
);

SELECT * FROM blood_bank_management_system.branch_phonenumbers_relation;
insert into Branch_Phonenumbers_Relation values('Br011','0112856947');
insert into Branch_Phonenumbers_Relation values('Br003','0112587364');
insert into Branch_Phonenumbers_Relation values('Br024','0112954126');
insert into Branch_Phonenumbers_Relation values('Br014','0112796458');
insert into Branch_Phonenumbers_Relation values('Br009','0112249637');
insert into Branch_Phonenumbers_Relation values('Br011','0112456286');
insert into Branch_Phonenumbers_Relation values('Br021','0112111597');
insert into Branch_Phonenumbers_Relation values('Br037','0112644799');
insert into Branch_Phonenumbers_Relation values('Br016','0112266657');

```

**Table: branch\_availablebloodgroups\_relation**

**Columns:**

- Branch\_ID** varchar(50) PK
- Available\_Blood\_Groups** varchar(50) PK
- Available\_Blood\_Quantity** int

| Branch_ID | Contact_No |
|-----------|------------|
| Br003     | 0112587364 |
| Br009     | 0112249637 |
| Br11      | 0112456286 |
| Br11      | 0112856947 |
| Br14      | 0112796458 |
| Br16      | 0112266657 |
| Br21      | 0112111597 |
| Br003     | 0112587364 |

**Object Info** **Session**

Query Completed

### 3.1.13. Donor table

```

insert into DONOR values('D651354','Dineth','2001-06-25','Male','Sinhalese',1,390,'2022-03-25','O+','Hilograma','kurunegala','North Western','B33467');
insert into DONOR values('D651355','Tharushi','2000-09-05','Female','Sinhalese',1,500,'2023-10-20','A+','Udasgiriya','Matale','Central','B33469');
insert into DONOR values('D651357','Kavya','2002-11-11','Female','Tamil',1,450,'2023-12-04','B+','Thirunagar North','Kilinochchi','Northern','B33470');
insert into DONOR values('D651360','Nimaya','2001-03-18','Female','Sinhalese',1,400,'2022-01-30','A-','Huriagawena','Anuradhapura','North Central','B33486');
insert into DONOR values('D651362','Yugan','2000-08-29','Male','Tamil',1,385,'2020-10-06','O-','Barathypuram','Kilinochchi','Northern','B33492');
insert into DONOR values('D651366','Aravind','2001-12-01','Male','Tamil',1,470,'2022-05-27','O+','Parantan-Mullaithivu Hwy','Tanniyuttu','Northern','B33493');
insert into DONOR values('D651369','Kavindu','2002-06-28','Male','Sinhalese',1,430,'2022-04-05','O-','Rupaha','Numara Eliya','Central','B33500');

```

**Schema: foe\_23**

| Donor_ID | Donor_Name | Date_of_Birth | Gender | Nationality | Eligibility_Status | Blood_Quantity | Last_Donated_Date | BloodGroup | Street                   | City         | Province      | bloodid |
|----------|------------|---------------|--------|-------------|--------------------|----------------|-------------------|------------|--------------------------|--------------|---------------|---------|
| D651354  | Dineth     | 2001-06-25    | Male   | Sinhalese   | 1                  | 390            | 2022-03-25        | O+         | Hilograma                | kurunegala   | North Western | B33467  |
| D651355  | Tharushi   | 2000-09-05    | Female | Sinhalese   | 1                  | 500            | 2023-10-20        | A+         | Udasgiriya               | Matale       | Central       | B33469  |
| D651357  | Kavya      | 2002-11-11    | Female | Tamil       | 1                  | 450            | 2023-12-04        | B+         | Thirunagar North         | Kilinochchi  | Northern      | B33470  |
| D651360  | Nimaya     | 2001-03-18    | Female | Sinhalese   | 1                  | 400            | 2022-01-30        | A-         | Huriagawena              | Anuradhapura | North Central | B33486  |
| D651362  | Yugan      | 2000-08-29    | Male   | Tamil       | 1                  | 385            | 2020-10-06        | O-         | Barathypuram             | Kilinochchi  | Northern      | B33492  |
| D651366  | Aravind    | 2001-12-01    | Male   | Tamil       | 1                  | 470            | 2022-05-27        | O+         | Parantan-Mullaithivu Hwy | Tanniyuttu   | Northern      | B33493  |
| D651369  | Kavindu    | 2002-06-28    | Male   | Sinhalese   | 1                  | 430            | 2022-04-05        | O-         | Rupaha                   | Numara Eliya | Central       | B33500  |

**Action Output**

| #   | Time     | Action  | Message | Duration / Fetch |
|-----|----------|---|---------|------------------|
| 117 | 20:48:19 | insert into BLOOD_DONATION_CAMP values('Br014','Camp4','2024-01-22','Street4','Kununegala', No ... 1 row(s) affected) |         | 0.000 sec        |

Object Info Session

Query Completed

### 3.1.14. Donor- Health Problems Table

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** Shows various tables like blood\_patient\_relation, bloodbank\_hospital\_relation, branch\_availablebloodgroup, branch\_phonenumbers\_relation, donor, donor\_blooddonationcamp\_, donor\_donateddates\_relation, donor\_email\_relation, donor\_healthproblems\_relation, donor\_phonenumbers\_relation, employee, employee\_donor\_relation, employee\_email\_relation, employee\_phonenumbers\_relation, equipment, hospital, hospital\_phonenumbers\_relation.
- Table: donor\_healthproblems\_relation**
- Columns:**
  - Donor\_ID** varchar(50) PK
  - Health\_Problems** varchar(250) PK
- Script:** The CREATE TABLE statement and several INSERT statements are shown in the SQL editor.
- Result Grid:** Displays the data inserted into the table:
 

| Donor_ID | Health_Problems |
|----------|-----------------|
| D651354  | Mild Anemia     |
| D651355  | none            |
| D651357  | Acne            |
| D651360  | Migraines       |
| D651362  | none            |
| D651366  | Hypertension    |
| D651369  | none            |
| NULL     | NULL            |

### 3.1.15. Donor- Donated Dates Table

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** Shows various tables like blood\_patient\_relation, bloodbank\_hospital\_relation, branch\_availablebloodgroup, branch\_phonenumbers\_relation, donor, donor\_blooddonationcamp\_, donor\_donateddates\_relation, donor\_email\_relation, donor\_healthproblems\_relation, donor\_phonenumbers\_relation, employee, employee\_donor\_relation, employee\_email\_relation, employee\_phonenumbers\_relation, equipment, hospital, hospital\_phonenumbers\_relation.
- Table: donor\_donateddates\_relation**
- Columns:**
  - Donor\_ID** varchar(50) PK
  - DonatedDate** varchar(50) PK
- Script:** The CREATE TABLE statement and several INSERT statements are shown in the SQL editor.
- Result Grid:** Displays the data inserted into the table:
 

| Donor_ID | DonatedDate |
|----------|-------------|
| D651354  | 2024-01-30  |
| D651355  | 2024-02-10  |
| D651357  | 2024-01-13  |
| D651360  | 2024-01-22  |
| D651362  | 2024-02-17  |
| D651366  | 2024-03-12  |
| D651369  | 2024-01-02  |
| NULL     | NULL        |

### 3.1.16. Donor- Email Table

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

**Table: donor\_email\_relation**

```

184 • CREATE TABLE Donor_Email_Relation(
185     Donor_ID varchar(50) not null,
186     Email varchar(50)not null,
187     primary key(Donor_ID,Email),
188     constraint fk_donor foreign key(Donor_ID) references DONOR(Donor_ID)
189     ON DELETE CASCADE ON UPDATE CASCADE
190   );
191   );
192 • SELECT * FROM blood_bank_management_system.donor_email_relation;
193 • insert into Donor_Email_Relation values('D651354','dineeth2001@gmail.com');
194 • insert into Donor_Email_Relation values('D651355','tharush2000@gmail.com');
195 • insert into Donor_Email_Relation values('D651357','kavya2002@gmail.com');
196 • insert into Donor_Email_Relation values('D651360','nimaya2001@gmail.com');
197 • insert into Donor_Email_Relation values('D651362','yugan2000@gmail.com');
198 • insert into Donor_Email_Relation values('D651366','aravind2001@gmail.com');
199 • insert into Donor_Email_Relation values('D651369','kavindu2002@gmail.com');
200
  
```

**Table: donor**

**Columns:**

| Donor_ID           | varchar(50) PK |
|--------------------|----------------|
| Donor_Name         | varchar(50)    |
| Date_of_Birth      | date           |
| Nationality        | varchar(40)    |
| Eligibility_Status | tinyint(1)     |
| Blood_Quantity     | int            |
| Last_Donated_Date  | date           |
| Blood_Group        | varchar(3)     |
| Street             | varchar(10)    |
| City               | varchar(50)    |
| Province           | varchar(50)    |
| Blood_ID           | varchar(40)    |

**Result Grid**

| Donor_ID | Email                 |
|----------|-----------------------|
| D651354  | dineeth2001@gmail.com |
| D651355  | tharush2000@gmail.com |
| D651357  | kavya2002@gmail.com   |
| D651360  | nimaya2001@gmail.com  |
| D651362  | yugan2000@gmail.com   |
| D651366  | aravind2001@gmail.com |
| D651369  | kavindu2002@gmail.com |

**donor\_email\_relation10**

**Output**

### 3.1.17. Donor- Phone Number Tables

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

**Table: donor\_phonenumbers\_relation**

```

218 • CREATE TABLE Donor_Phonenumbers_Relation(
219     Donor_ID varchar(50) not null ,
220     Contact_No varchar(10) not null,
221     primary key(Donor_ID,Contact_No),
222     constraint fk_donor1 foreign key(Donor_ID) references DONOR(Donor_ID)
223     ON DELETE CASCADE ON UPDATE CASCADE
224   );
225
226 • insert into Donor_Phonenumbers_Relation values('D651354','0771256948');
227 • insert into Donor_Phonenumbers_Relation values('D651355','0704659832');
228 • insert into Donor_Phonenumbers_Relation values('D651357','0784596327');
229 • insert into Donor_Phonenumbers_Relation values('D651360','0777458620');
230 • insert into Donor_Phonenumbers_Relation values('D651362','0704569881');
231 • insert into Donor_Phonenumbers_Relation values('D651366','0712582588');
232 • insert into Donor_Phonenumbers_Relation values('D651369','0712546999');
233
  
```

**Table: donor\_donateddates\_relation**

**Columns:**

| Donor_ID    | varchar(50) PK |
|-------------|----------------|
| DonatedDate | varchar(50) PK |

**Result Grid**

| Donor_ID | Contact_No |
|----------|------------|
| D651354  | 0771256948 |
| D651355  | 0704659832 |
| D651357  | 0784596327 |
| D651360  | 0777458620 |
| D651362  | 0704569881 |
| D651366  | 0712582588 |
| D651369  | 0712546999 |

**donor\_phonenumbers\_relation12**

**Output**

### **3.1.18. Donor- Blood Donation Camp Table**

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas, Administration, Schemas (selected), Information.
- Central Area:** Title: Blood Bank Management System. A code editor window displays the SQL script for creating the `Donor_BloodDonationCamp_Relation` table and inserting data. The table has columns: `donor_id`, `branch_id`, `camp_name`, and `held_date`. Primary key constraints are defined on `donor_id` and `branch_id, camp_name, held_date`. Foreign key constraints link `donor_id` to `DONOR(Donor_ID)` and `branch_id, camp_name, held_date` to `BLOOD_DONATION_CAMP(Branch_ID, Camp_Name, Held_Date)`. The script also includes a comment `-- DONE`.
- Result Grid:** Shows the inserted data in a grid format:

| donor_id | branch_id | camp_name | held_date  |
|----------|-----------|-----------|------------|
| D651355  | Br003     | Camp2     | 2024-02-10 |
| D651354  | Br011     | Camp1     | 2024-01-30 |
| D651363  | Br014     | Camp4     | 2024-01-22 |
| D651369  | Br016     | Camp8     | 2024-01-02 |
| D651362  | Br021     | Camp6     | 2024-03-12 |
| D651357  | Br024     | Camp3     | 2024-01-13 |
| D651366  | Br037     | Camp7     | 2024-02-20 |
|          |           |           |            |

- Bottom Navigation:** Object Info, Session, Output, Apply, Revert.

### 3.1.19. Employee Table

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Local instance MySQL80 (blo...).
- Navigator:** Schemas (employee).
- Table Editor:** Employee table definition:

```
CREATE TABLE EMPLOYEE(
    Employee_ID varchar(40) not null,
    Employee_Name varchar(40) not null,
    Gender varchar(10) not null,
    Nationality varchar(40) not null,
    Employee_Type varchar(40) not null,
    Date_of_Birth date not null,
    Salary int default 0,
    Street varchar(40) not null,
    City varchar(40) not null,
    Province varchar(40) not null,
    primary key(Employee_ID),
    unique key(Employee_Name,Employee_Type,Date_of_Birth),
    Supervisor_ID varchar(40),
    Branch_ID varchar(40) not null,
    -- constraint fk_emp_fk_branch foreign key(Branch_ID) references BRANCH(Branch_ID)
    constraint fk_bloodbank3 foreign key(Branch_ID) references BRANCH(Branch_ID)
)
```
- Result Grid:** Employee table data:

| Employee_ID | Employee_Name | Gender | Nationality | Employee_Type                   | Date_of_Birth | Salary | Street   | City         | Province      | Supervisor_ID | Branch_ID |
|-------------|---------------|--------|-------------|---------------------------------|---------------|--------|----------|--------------|---------------|---------------|-----------|
| E008        | Chamila       | Female | Sinhalese   | Customer Service Representative | 1985-03-21    | 40000  | Street33 | Badulla      | Uva           | S42           | B009      |
| E034        | Kamal         | Male   | Sinhalese   | Administrative Staff            | 1990-04-29    | 50000  | Street63 | Anuradhapura | North Central | S24           | B011      |
| E041        | Nimal         | Female | Sinhalese   | Nurse                           | 1985-09-21    | 80000  | Street47 | Ampara       | Eastern       | S32           | B003      |
| E045        | Keerthis      | Male   | Sinhalese   | Nurse                           | 1983-09-21    | 80000  | Street60 | Kurunegala   | North Western | S41           | B014      |
| E053        | Kanthi        | Female | Tamil       | Information Technology Support  | 1991-10-11    | 100000 | Street50 | Jaffna       | Northern      | S34           | B024      |
| E058        | Sunil         | Male   | Sinhalese   | Nurse                           | 1985-07-23    | 90000  | Street74 | Colombo      | Western       | S56           | B016      |
| E066        | Ranjith       | Male   | Sinhalese   | Research Scientist              | 1968-10-27    | 150000 | Street32 | Kandy        | Central       | S52           | B037      |
| E072        | Pradeep       | Male   | Sinhalese   | Administrative Staff            | 1984-09-21    | 60000  | Street51 | Peradeniya   | Central       | S47           | B021      |
| E073        | Shyama        | Female | Sinhalese   | Nurse                           | 1982-12-21    | 80000  | Street55 | Peradeniya   | Central       | S48           | B021      |
|             |               |        |             |                                 |               |        |          |              |               |               |           |
|             |               |        |             |                                 |               |        |          |              |               |               |           |
- Object Info:** Employee table.
- Output:** Query Completed.

### 3.1.20. Employee- Phone Numbers

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** Blood Bank Management System.
- Code Editor:** Shows the SQL code for creating the `Employee_PhoneNumbers_Relation` table and inserting data into it. The table has columns `Employee_ID` and `Contact_No`. Primary key constraint is defined on `(Employee_ID, Contact_No)`. Foreign key constraint `fk_empl` references the `EMPLOYEE` table's `Employee_ID`.
- Result Grid:** Displays the inserted data, showing rows for employees E008, E034, E041, E045, E053, E058, E066, E072, E073, and E074.
- Table:** `employee` (selected in the left sidebar). It has columns: `Employee_ID` (PK), `Employee_Name`, `Gender`, `Nationality`, `Employee_Type`, `Date_of_Birth`, `Salary`, `Street`, `City`, `Province`, `Supervisor_ID`, and `Branch_ID`.
- Information:** Shows the `employee_phonenumbers_relation` table.

### 3.1.21. Employee- Donor Table Relation

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** Blood Bank Management System.
- Code Editor:** Shows the SQL code for creating the `Employee_Donor_Relation` table and inserting data into it. The table has columns `donor_id` and `employee_id`. Primary key constraint is defined on `(donor_id, employee_id)`. Foreign key constraints `fk_donor3` and `fk_empl3` reference the `DONOR` and `EMPLOYEE` tables respectively.
- Result Grid:** Displays the inserted data, showing rows for donors D651360, D651354, D651355, D651357, D651369, D651366, D651362, and D651367.
- Table:** `employee_donor_relation` (selected in the left sidebar). It has columns: `donor_id` (PK) and `employee_id`.
- Information:** Shows the `employee_donor_relation` table.

### 3.1.22. Employee- Email Table

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** blood\_bank\_management\_system
- Table:** employee\_email\_relation
- Columns:**
  - Employee\_ID: varchar(40) PK
  - Email: varchar(40)
- Code:** SQL script for creating the table and inserting data.
- Result Grid:** Shows the inserted data:

| Employee_ID | Email             |
|-------------|-------------------|
| E008        | chanila@gmail.com |
| E034        | kamal@gmail.com   |
| E041        | nimali@gmail.com  |
| E045        | kamal@gmail.com   |
| E053        | keerthi@gmail.com |
| E058        | sunil@gmail.com   |
| E066        | ranjith@gmail.com |
| E072        | pradeep@gmail.com |
| E073        | shyama@gmail.com  |
| E066        | ranjith@gmail.com |
| E072        | pradeep@gmail.com |
| E073        | shyama@gmail.com  |

### 3.1.23. Equipment Table

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** blood\_bank\_management\_system
- Table:** EQUIPMENT
- Columns:**
  - Eqip\_Name: varchar(50) not null
  - Cost: integer default 0
  - Quantity: integer default 0
  - Equip\_Description: varchar(250)
  - Branch\_ID: varchar(50) not null
- Code:** SQL script for creating the table and inserting data.
- Result Grid:** Shows the inserted data:

| blood_id | Eqip_Name                | Branch_ID |
|----------|--------------------------|-----------|
| B33467   | Blood bank refrigerator  | Br003     |
| B33467   | Blood collection monitor | Br011     |
| B33469   | Blood bag tube sealer    | Br014     |
| B33469   | Blood donor couch        | Br024     |
| B33469   | Blood mixer roller       | Br009     |

### 3.1.24. Hospital Table

MySQL Workbench

Local instance MySQL80 (blo...)

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

**Table: hospital**

**Columns:**

- Hospital\_ID varchar(50) PK
- Hospital\_Name varchar(100)
- Street varchar(50)
- City varchar(50)
- Province varchar(50)

Information

Result Grid

```

CREATE TABLE HOSPITAL(
    Hospital_ID varchar(50)not null,
    Hospital_Name varchar(100) not null,
    Street varchar(50) not null,
    City varchar(50) not null,
    Province varchar(50) not null,
    PRIMARY KEY (Hospital_ID)
);

insert into HOSPITAL values('H008','General Hospital','StreetH1','Kandy','Central');
insert into HOSPITAL values('H010','Co-operative Hospital','StreetH2','Matale','Central');
insert into HOSPITAL values('H011','Kantalai Base Hospital','StreetH3','Trincomalee','Eastern');
insert into HOSPITAL values('H015','Kurunegala Hospital','StreetH4','Kurunegala','North Western');
insert into HOSPITAL values('H016','Chilaw General Hospital','StreetH5','Puttalam','North Western');
insert into HOSPITAL values('H019','Rambukkana District Hospital','StreetH6','Ratnapura','Sabaragamuwa');
insert into HOSPITAL values('H022','Karapitiya teaching Hospital','StreetH7','Galle','Southern');

```

Object Info Session Output

Query Completed

### 3.1.25. Hospital- Phone Numbers Table

MySQL Workbench

Local instance MySQL80 (blo...)

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

**Table: hospital**

**Columns:**

- Hospital\_ID varchar(50) PK
- Hospital\_Name varchar(100)
- Street varchar(50)
- City varchar(50)
- Province varchar(50)

Information

Result Grid

```

CREATE TABLE Hospital_PhoneNumbers_Relation(
    Hospital_ID varchar(50) not null,
    Contact_No varchar(10) not null,
    primary key(Hospital_ID,Contact_No),
    constraint fk_hospital foreign key(Hospital_ID) references HOSPITAL(Hospital_ID)
);

SELECT * FROM blood_bank_management_system.hospital_phonenumbers_relation;
insert into Hospital_PhoneNumbers_Relation values('H008','0112345876');
insert into Hospital_PhoneNumbers_Relation values('H010','0112334759');
insert into Hospital_PhoneNumbers_Relation values('H011','0112887955');
insert into Hospital_PhoneNumbers_Relation values('H015','0112495711');
insert into Hospital_PhoneNumbers_Relation values('H016','0112151484');
insert into Hospital_PhoneNumbers_Relation values('H019','0112622522');
insert into Hospital_PhoneNumbers_Relation values('H022','0112860960');

```

Object Info Session Output

Query Completed

### 3.1.26. Patient Table

MySQL Workbench

Local instance MySQL80 (blo...)

File Edit View Query Database Server Tools Scripting Help

Navigator

**SCHEMAS**

Table: **employee\_email\_relation**

Columns:

- Employee\_ID** varchar(40) PK
- Email varchar(40)

Object Info Session

Query Completed

Blood Bank Management System...

```

337 • CREATE TABLE PATIENT(
338     Patient_ID varchar(40) not null,
339     Patient_Name varchar(40) not null,
340     Gender varchar(10) not null,
341     Nationality varchar(40) not null,
342     Date_of_Birth date not null,
343     Street varchar(40) not null,
344     City varchar(40) not null,
345     Province varchar(40) not null,
346     primary key(Patient_ID),
347     unique key(Patient_Name,Date_of_Birth)
348 );
349
350 • insert into PATIENT values('P23487','Ravindu','Male','Sinhalese','2000-06-24','Street64','Kaduwela','Western');
351 • insert into PATIENT values('P23488','Kavindu','Male','Sinhalese','1995-11-09','Street81','Bandarawela','Uva');
352 • insert into PATIENT values('P23489','Rani','Female','Sinhalese','1976-03-21','Street57','Hatton','Central');
353 • insert into PATIENT values('P23490','Pramitha','Male','Tamil','1970-12-24','Street32','Chavakachcheri','Northern');
354 • insert into PATIENT values('P23491','Samitha','Female','Sinhalese','2001-01-30','Street84','Peliyagoda','Western');

```

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

| Patient_ID | Patient_Name | Gender | Nationality | Date_of_Birth | Street   | City           | Province |
|------------|--------------|--------|-------------|---------------|----------|----------------|----------|
| P23487     | Ravindu      | Male   | Sinhalese   | 2000-06-24    | Street64 | Kaduwela       | Western  |
| P23488     | Kavindu      | Male   | Sinhalese   | 1995-11-09    | Street81 | Bandarawela    | Uva      |
| P23489     | Rani         | Female | Sinhalese   | 1976-03-21    | Street57 | Hatton         | Central  |
| P23490     | Pramitha     | Male   | Tamil       | 1970-12-24    | Street32 | Chavakachcheri | Northern |
| P23491     | Samitha      | Female | Sinhalese   | 2001-01-30    | Street84 | Peliyagoda     | Western  |
| NULL       | NULL         | NULL   | NULL        | NULL          | NULL     | NULL           | NULL     |

patient 18 x

Output

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

| Patient_ID | Contact_No |
|------------|------------|
| P23487     | 0759834571 |
| P23488     | 0714544841 |
| P23489     | 0772582588 |
| P23490     | 0779115112 |
| P23491     | 0701477899 |
| NULL       | NULL       |

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

### 3.1.27. Patient- Phone Numbers Table

MySQL Workbench

Local instance MySQL80 (blo...)

File Edit View Query Database Server Tools Scripting Help

Navigator

**SCHEMAS**

Table: **employee\_email\_relation**

Columns:

- Employee\_ID** varchar(40) PK
- Email varchar(40)

Object Info Session

Query Completed

Blood Bank Management System...

```

355 • -- create table Patient_PhoneNumbers_Relation(
356     Patient_ID varchar(40) not null,
357     Contact_No varchar(10) not null,
358     primary key(Patient_ID,Contact_No),
359     constraint fk_patient foreign key(Patient_ID) references PATIENT(Patient_ID)
360     ON DELETE CASCADE ON UPDATE CASCADE
361 );
362
363 • SELECT * FROM blood_bank_management_system.patient_phonenumbers_relation;
364 • insert into Patient_PhoneNumbers_Relation values('P23487','0759834571');
365 • insert into Patient_PhoneNumbers_Relation values('P23488','0714544841');
366 • insert into Patient_PhoneNumbers_Relation values('P23489','0772582588');
367 • insert into Patient_PhoneNumbers_Relation values('P23490','0779115112');
368 • insert into Patient_PhoneNumbers_Relation values('P23491','0701477899');

```

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

| Patient_ID | Contact_No |
|------------|------------|
| P23487     | 0759834571 |
| P23488     | 0714544841 |
| P23489     | 0772582588 |
| P23490     | 0779115112 |
| P23491     | 0701477899 |
| NULL       | NULL       |

patient\_phonenumbers\_relation...

Output

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

### **3.1.28. Patient- Email Table**

MySQL Workbench

Local instance MySQL80 (blo...)

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- branch\_availablebloodgroup
- branch\_phonenumbers\_relati...
- donor
- donor\_blooddonationcamp...
- donor\_donateddates\_relati...
- donor\_email\_relation
- donor\_healthproblems\_relat...
- donor\_phonenumbers\_relati...
- employee
- employee\_donor\_relation
- employee\_email\_relation
- employee\_phonenumbers\_rela...
- equipment
- hospital
- hospital\_phonenumbers\_rela...
- patient
- patient\_email\_relation
- patient\_healthproblems\_rela...

Administration Schemas

Information

Table: patient\_email\_relation

Columns:

|            |                |
|------------|----------------|
| Patient_ID | varchar(40) PK |
| Email      | varchar(40) PK |

Blood Bank Management System...

Limit to 1000 rows

```
370 • create table Patient_Email_Relation(
371     Patient_ID varchar(40) not null,
372     Email varchar(40) not null,
373     primary key(Patient_ID,Email),
374     constraint fk_patient1 foreign key(Patient_ID) references PATIENT(Patient_ID)
375         ON DELETE CASCADE ON UPDATE CASCADE
376 );
377
378 • insert into Patient_Email_Relation values('P23487','ravindu@gmail.com');
379 • insert into Patient_Email_Relation values('P23488','kavindu@gmail.com');
380 • insert into Patient_Email_Relation values('P23489','rani@gmail.com');
381 • insert into Patient_Email_Relation values('P23490','pramitha@gmail.com');
382 • insert into Patient_Email_Relation values('P23491','samitha@gmail.com');
383
```

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

| Patient_ID | Email              |
|------------|--------------------|
| P23487     | ravindu@gmail.com  |
| P23488     | kavindu@gmail.com  |
| P23489     | rani@gmail.com     |
| P23490     | pramitha@gmail.com |
| P23491     | samitha@gmail.com  |
| *          | HULL               |

Result Grid | Form Editor | Field Types | Query Stats | Object Info | Session | Output | Apply | Revert

patient\_email\_relation 20 x

Query Completed

### **3.1.29. Patient- Health Problems Table**

### 3.1.30. Payment Table

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** The current schema is "Blood Bank Management System".
- Code Editor:** The code being run is the creation of the PAYMENT table and its initial data insertion. The table has columns: Payment\_ID (varchar(50) not null), Amount (decimal(3,1) not null), Paid\_Date (date not null), and Patient\_ID (varchar(50) not null). It includes a primary key constraint on Payment\_ID and a foreign key constraint fk\_patient3 referencing the PATIENT table's Patient\_ID.
- Result Grid:** The result grid shows the inserted data for the PAYMENT table. The first five rows are:
 

| Payment_ID | Amount | Paid_Date  | Patient_ID |
|------------|--------|------------|------------|
| PID8759284 | 20.0   | 2024-03-09 | P23491     |
| PID8759297 | 60.0   | 2024-03-12 | P23489     |
| PID8759301 | 25.0   | 2024-03-15 | P23488     |
| PID8759342 | 20.0   | 2024-02-27 | P23490     |

 There are two more rows with NULL values for Patient\_ID.
- Information:** A table named patient\_healthproblems\_relation is shown with the following columns:
 

| Patient_ID | Health_Problems |
|------------|-----------------|
|------------|-----------------|

 Both columns are defined as varchar(40) and PK.
- Object Info:** Shows the status of the object as "Query Completed".

## 3.3. Table Alteration & Deriving Age from Date of Birth

### 3.3.1. Donor Table

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** The current schema is "Blood Bank Management System".
- Code Editor:** The code runs several ALTER TABLE statements to add new columns to the DONOR, EMPLOYEE, and PATIENT tables. It also updates the EMPLOYEE and PATIENT tables to set the new Age column based on the timestamp difference between their birth date and the current date.
- Result Grid:** The result grid shows the updated data for the PATIENT table. The first few rows are:
 

| Donor_ID | Donor_Name | Date_of_Birth | Nationality | Eligibility_Status | Blood_Quantity | Last_Donated_Date | Blood_Group | Street           | City        | Province      | Blood_ID | Age | Gender |
|----------|------------|---------------|-------------|--------------------|----------------|-------------------|-------------|------------------|-------------|---------------|----------|-----|--------|
| D651354  | Dineth     | 2001-06-25    | Sinhalese   | 1                  | 390            | 2022-03-25        | O+          | Hilogama         | kurunegala  | North Western | B33467   | 22  | NULL   |
| D651355  | Tharushi   | 2000-09-05    | Sinhalese   | 1                  | 500            | 2023-10-20        | A+          | Udagiriya        | Matale      | Central       | B33469   | 23  | NULL   |
| D651357  | Kavya      | 2002-11-11    | Tamil       | 1                  | 450            | 2023-12-04        | B+          | Thirunagar North | Kilinochchi | Northern      | B33470   | 21  | NULL   |

 There are more rows with various data.
- Information:** A table named donor is shown with the following columns:
 

| Donor_ID | Donor_Name | Date_of_Birth | Nationality | Eligibility_Status | Blood_Quantity | Last_Donated_Date | Blood_Group | Street | City | Province | Blood_ID | Age | Gender |
|----------|------------|---------------|-------------|--------------------|----------------|-------------------|-------------|--------|------|----------|----------|-----|--------|
|----------|------------|---------------|-------------|--------------------|----------------|-------------------|-------------|--------|------|----------|----------|-----|--------|

 All columns are defined as varchar(40) except for Age and Gender which are int.
- Object Info:** Shows the status of the object as "Query Completed".

### 3.3.2. Patient Table

The screenshot shows the MySQL Workbench interface with the 'patient' table selected. The left sidebar displays the schema structure, including tables like donor\_phonenumbers\_relation, employee, and patient. The main pane shows the SQL history with several ALTER TABLE and UPDATE statements, followed by the table definition. The result grid displays the following data:

| Patient_ID | Patient_Name | Gender | Nationality | Date_of_Birth | Street   | City           | Province | Age  |
|------------|--------------|--------|-------------|---------------|----------|----------------|----------|------|
| P23487     | Ravindu      | Male   | Sinhalese   | 2000-06-24    | Street64 | Kaduwela       | Western  | 23   |
| P23488     | Kavindu      | Male   | Sinhalese   | 1995-11-09    | Street81 | Bandarawela    | Uva      | 28   |
| P23489     | Rani         | Female | Sinhalese   | 1976-03-21    | Street57 | Hatton         | Central  | 48   |
| P23490     | Pramitha     | Male   | Tamil       | 1970-12-24    | Street32 | Chavakkadcheri | Northern | 53   |
| P23491     | Samitha      | Female | Sinhalese   | 2061-01-30    | Street84 | Peliyagoda     | Western  | -36  |
| NULL       | NULL         | NULL   | NULL        | NULL          | NULL     | NULL           | NULL     | NULL |

## 3.4. Table Definitions

### 3.4.1. Blood Table

The screenshot shows the MySQL Workbench interface with the 'patient' table selected. The left sidebar displays the schema structure. The main pane shows the SQL history with several SHOW CREATE TABLE statements for various tables, followed by the creation of the 'blood' table. A modal window titled 'Edit Data for Create Table (VARCHAR)' shows the table definition:

```
CREATE TABLE `blood` (
  `Blood_ID` varchar(40) NOT NULL,
  `Blood_Group` varchar(3) NOT NULL,
  `Expiry_Date` date NOT NULL,
  `Blood_Quantity_ml` int DEFAULT '0',
  `Blood_Description` longtext,
  PRIMARY KEY (`Blood_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

The result grid shows the creation command:

| Table | Create Table   |
|-------|--|
| blood | CREATE TABLE `blood` ( `Blood_ID` varchar(40) NOT NULL, `Blood_Group` varchar(3) NOT NULL, `Expiry_Date` date NOT NULL, `Blood_Quantity_ml` int DEFAULT '0', `Blood_Description` longtext, PRIMARY KEY (`Blood_ID`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |

### 3.4.2. Blood- Branch Relation Table

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. In the 'Navigator' pane, the 'Tables' section is expanded, showing various tables like donor, patient, and blood\_branch\_relation. The 'Table: patient' section is selected, displaying its columns: Patient\_ID (PK), Patient\_Name, Gender, Nationality, Date\_of\_Birth, Street, City, and Province. A context menu is open over the 'blood\_branch\_relation' table, and a sub-menu window titled 'Edit Data for Create Table (VARCHAR)' is displayed. It contains the SQL code for creating the table:

```
CREATE TABLE `blood_branch_relation` (
  `blood_id` varchar(40) NOT NULL,
  `branch_id` varchar(50) NOT NULL,
  PRIMARY KEY (`blood_id`, `branch_id`),
  KEY `fk_branch1` (`branch_id`),
  CONSTRAINT `fk_blood3` FOREIGN KEY (`blood_id`) REFERENCES `blood` (`Blood_ID`) ON DELETE CASCADE
  ON UPDATE CASCADE,
  CONSTRAINT `fk_branch1` FOREIGN KEY (`branch_id`) REFERENCES `blood_bank` (`Branch_ID`) ON DELETE CASCADE
  ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

The 'Result Grid' shows the CREATE TABLE command. The status bar at the bottom right indicates '8:43 PM 3/31/2024'.

### 3.4.3. Blood - Blood Donation Camps Table

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. In the 'Navigator' pane, the 'Tables' section is expanded, showing various tables like donor, patient, and blood\_branch\_relation. The 'Table: patient' section is selected, displaying its columns: Patient\_ID (PK), Patient\_Name, Gender, Nationality, Date\_of\_Birth, Street, City, and Province. A context menu is open over the 'blood\_donation\_camp' table, and a sub-menu window titled 'Edit Data for Create Table (VARCHAR)' is displayed. It contains the SQL code for creating the table:

```
CREATE TABLE `blood_donation_camp` (
  `Branch_ID` varchar(40) NOT NULL,
  `Camp_Name` varchar(50) NOT NULL,
  `Held_Date` date NOT NULL,
  `Street` varchar(100) NOT NULL,
  `City` varchar(50) NOT NULL,
  `Province` varchar(50) NOT NULL,
  PRIMARY KEY (`Branch_ID`, `Camp_Name`, `Held_Date`),
  CONSTRAINT `fk_branch` FOREIGN KEY (`Branch_ID`) REFERENCES `blood_bank` (`Branch_ID`) ON DELETE CASCADE
  ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

The 'Result Grid' shows the CREATE TABLE command. The status bar at the bottom right indicates '8:43 PM 3/31/2024'.

### 3.4.4. Blood- Employee Relation Table

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. In the 'Schemas' tree, the 'patient' schema is expanded, showing its tables: donor\_phonenumbers\_relation, employee, employee\_donor\_relation, employee\_email\_relation, employee\_phonenumbers\_re, equipment, hospital, hospital\_phonenumbers\_rel, patient, patient\_email\_relation, patient\_healthproblems\_rela, patient\_phonenumbers\_rel, and payment. A context menu is open over the 'patient' table, and a sub-menu window titled 'Edit Data for Create Table (VARCHAR)' is displayed. It contains the SQL code for creating the 'blood\_employee\_relation' table:

```
CREATE TABLE `blood_employee_relation` (
  `blood_id` varchar(40) NOT NULL,
  `employee_id` varchar(50) NOT NULL,
  PRIMARY KEY (`blood_id`, `employee_id`),
  KEY `fk_emp1` (`employee_id`),
  CONSTRAINT `fk_blood4` FOREIGN KEY (`blood_id`) REFERENCES `blood` (`Blood_ID`) ON DELETE CASCADE,
  CONSTRAINT `fk_empl2` FOREIGN KEY (`employee_id`) REFERENCES `employee` (`Employee_ID`) ON DELETE CASCADE ON UPDATE CASCADE,
  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

The 'Result Grid' shows the command: CREATE TABLE `blood\_employee\_relation` ( ... ). The status bar at the bottom right indicates '8:43 PM 3/31/2024'.

### 3.4.5. Blood- Equipment Relation Table

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. In the 'Schemas' tree, the 'patient' schema is expanded, showing its tables: donor\_phonenumbers\_relation, employee, employee\_donor\_relation, employee\_email\_relation, employee\_phonenumbers\_re, equipment, hospital, hospital\_phonenumbers\_rel, patient, patient\_email\_relation, patient\_healthproblems\_rela, patient\_phonenumbers\_rel, and payment. A context menu is open over the 'patient' table, and a sub-menu window titled 'Edit Data for Create Table (VARCHAR)' is displayed. It contains the SQL code for creating the 'blood\_equipment\_relation' table:

```
CREATE TABLE `blood_equipment_relation` (
  `blood_id` varchar(40) NOT NULL,
  `Eqip_Name` varchar(50) NOT NULL,
  `Branch_ID` varchar(50) NOT NULL,
  PRIMARY KEY (`Eqip_Name`, `Branch_ID`, `blood_id`),
  KEY `fk_blood2` (`blood_id`),
  KEY `fk_branch2` (`Branch_ID`),
  CONSTRAINT `fk_blood2` FOREIGN KEY (`blood_id`) REFERENCES `blood` (`Blood_ID`) ON DELETE CASCADE,
  CONSTRAINT `fk_branch2` FOREIGN KEY (`Branch_ID`) REFERENCES `equipment` (`Branch_ID`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `fk_eqipment` FOREIGN KEY (`Eqip_Name`) REFERENCES `equipment` (`Eqip_Name`) ON DELETE CASCADE ON UPDATE CASCADE
  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

The 'Result Grid' shows the command: CREATE TABLE `blood\_equipment\_relation` ( ... ). The status bar at the bottom right indicates '8:44 PM 3/31/2024'.

### 3.4.6. Blood- Patient Table

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. In the center, a query editor window displays the SQL code for creating the 'blood\_patient\_relation' table. The code is as follows:

```
CREATE TABLE `blood_patient_relation` (
  `blood_id` varchar(40) NOT NULL,
  `patient_id` varchar(50) NOT NULL,
  PRIMARY KEY (`blood_id`, `patient_id`),
  KEY `fk_patient4` (`patient_id`),
  CONSTRAINT `fk_blood5` FOREIGN KEY (`blood_id`) REFERENCES `blood` (`Blood_ID`) ON DELETE CASCADE,
  CONSTRAINT `fk_patient4` FOREIGN KEY (`patient_id`) REFERENCES `patient` (`Patient_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

The 'Result Grid' tab shows the CREATE TABLE command. The status bar at the bottom right indicates '8:44 PM 3/31/2024'. The taskbar at the bottom shows various application icons.

### 3.4.7. Blood- Test Samples Relation Table

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. In the center, a query editor window displays the SQL code for creating the 'blood\_bloodtestsamples\_relation' table. The code is as follows:

```
CREATE TABLE `blood_bloodtestsamples_relation` (
  `Blood_ID` varchar(40) NOT NULL,
  `Test_Done_For_Blood_Samples` varchar(250) NOT NULL,
  PRIMARY KEY (`Blood_ID`, `Test_Done_For_Blood_Samples`),
  CONSTRAINT `fk_blood` FOREIGN KEY (`Blood_ID`) REFERENCES `blood` (`Blood_ID`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

The 'Result Grid' tab shows the CREATE TABLE command. The status bar at the bottom right indicates '8:42 PM 3/31/2024'. The taskbar at the bottom shows various application icons.

### 3.4.8. Blood Bank Table

The screenshot shows the MySQL Workbench interface with the 'Schemas' tab selected. A code editor window displays the following SQL script:

```
568 /* ***** Table Definitions ***** */
569 
570 • show create table blood;
571 • show create table blood_bank;
572 • show create table blood_bloodtestsamples_relation;
573 • show create table blood_branch_relation;
574 • show create table blood_donation_camp;
575 • show create table blood_employee_relation;
576 • show create table blood_patient_relation;
577 • show create table blood_equipment_relation;
578 • show create table bloodbank_hospital_relation;
579 • show create table branch_availablebloodgroups_relation;
580 • show create table branch_phonenumbers_relation;
581 • show create table donor;
582 • show create table donor_blooddonationcamp_relation;
583 • show create table donor_donateddates_relation;
584 • show create table donor_email_relation;
```

A modal dialog titled "Edit Data for Create Table (VARCHAR)" is open, showing the CREATE TABLE statement for the "blood\_bank" table:

```
1 CREATE TABLE `blood_bank` (
2   `Branch_ID` varchar(40) NOT NULL,
3   `Branch_Name` varchar(50) NOT NULL,
4   `Available_Blood_Quantity` int DEFAULT '0',
5   `Street` varchar(50) NOT NULL,
6   `City` varchar(50) NOT NULL,
7   `Province` varchar(50) NOT NULL,
8   PRIMARY KEY (`Branch_ID`)
9 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

The "Result Grid" shows the command: `CREATE TABLE `blood_bank` (...)`.

### 3.4.9. Blood Bank- Hospital Relation Table

The screenshot shows the MySQL Workbench interface with the 'Schemas' tab selected. A code editor window displays the following SQL script:

```
568 /* ***** Table Definitions ***** */
569 
570 • show create table blood;
571 • show create table blood_bank;
572 • show create table blood_bloodtestsamples_relation;
573 • show create table blood_branch_relation;
574 • show create table blood_donation_camp;
575 • show create table blood_employee_relation;
576 • show create table blood_patient_relation;
577 • show create table blood_equipment_relation;
578 • show create table blood_patient_relation;
579 
580 • show create table bloodbank_hospital_relation;
581 • show create table branch_availablebloodgroups_relation;
582 • show create table branch_phonenumbers_relation;
583 • show create table donor;
584 • show create table donor_blooddonationcamp_relation;
585 • show create table donor_donateddates relation;
```

A modal dialog titled "Edit Data for Create Table (VARCHAR)" is open, showing the CREATE TABLE statement for the "bloodbank\_hospital\_relation" table:

```
1 CREATE TABLE `bloodbank_hospital_relation` (
2   `hospital_id` varchar(40) NOT NULL,
3   `branch_id` varchar(50) NOT NULL,
4   PRIMARY KEY (`hospital_id`, `branch_id`),
5   KEY `fk_branch7` (`branch_id`),
6   CONSTRAINT `fk_branch7` FOREIGN KEY (`branch_id`) REFERENCES `blood_bank` (`Branch_ID`) ON DELETE CASCADE ON UPDATE CASCADE,
7   CONSTRAINT `fk_hospital2` FOREIGN KEY (`hospital_id`) REFERENCES `hospital` (`Hospital_ID`) ON DELETE CASCADE ON UPDATE CASCADE
8 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

The "Result Grid" shows the command: `CREATE TABLE `bloodbank_hospital_relation` (...)`.

### 3.4.10. Branch- Available Blood Groups Table

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. In the central pane, the code editor displays the following SQL script:

```
568 /* ***** Table Definitions ***** */
569 
570 • show create table blood;
571 • show create table blood_bank;
572 • show create table blood_bloodtestsamples_relation;
573 • show create table blood_branch_relation;
574 • show create table blood_donation_camp;
575 • show create table blood_employee_relation;
576 • show create table blood_equipment_relation;
577 • show create table blood_patient_relation;
578 • show create table bloodbank_hospital_relation;
579 • show create table branch_availablebloodgroups_relation;
580 
581 • show create table branch_phonenumbers_relation;
582 • show create table branch_phonenumbers_relation;
583 • show create table donor;
584 • show create table donor_blooddonationcamp_relation;
585 • show create table donor_donateddates relation;
```

The right-hand panel shows the 'Edit Data for Create Table (VARCHAR)' dialog for the 'branch\_availablebloodgroups\_relation' table. The table definition is:

```
CREATE TABLE `branch_availablebloodgroups_relation` (
  `Branch_ID` varchar(50) NOT NULL,
  `Available_Blood_Groups` varchar(3) NOT NULL,
  `Available_Blood_Quantity` int DEFAULT '0',
  PRIMARY KEY (`Branch_ID`, `Available_Blood_Groups`),
  CONSTRAINT `fk_bloodbank` FOREIGN KEY (`Branch_ID`) REFERENCES `blood_bank` (`Branch_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

The 'Result Grid' shows the command: CREATE TABLE `branch\_availablebloodgroups\_relation` ...

### 3.4.11. Branch- Phone Numbers Table

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. In the central pane, the code editor displays the following SQL script:

```
568 /* ***** Table Definitions ***** */
569 
570 • show create table blood;
571 • show create table blood_bank;
572 • show create table blood_bloodtestsamples_relation;
573 • show create table blood_branch_relation;
574 • show create table blood_donation_camp;
575 • show create table blood_employee_relation;
576 • show create table blood_equipment_relation;
577 • show create table blood_patient_relation;
578 • show create table bloodbank_hospital_relation;
579 • show create table branch_availablebloodgroups_relation;
580 
581 • show create table branch_phonenumbers_relation;
582 • show create table donor;
583 • show create table donor_blooddonationcamp_relation;
584 • show create table donor_donateddates relation;
```

The right-hand panel shows the 'Edit Data for Create Table (VARCHAR)' dialog for the 'branch\_phonenumbers\_relation' table. The table definition is:

```
CREATE TABLE `branch_phonenumbers_relation` (
  `Branch_ID` varchar(50) NOT NULL,
  `Contact_No` varchar(10) NOT NULL,
  PRIMARY KEY (`Branch_ID`, `Contact_No`),
  CONSTRAINT `fk_bloodbank1` FOREIGN KEY (`Branch_ID`) REFERENCES `blood_bank` (`Branch_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

The 'Result Grid' shows the command: CREATE TABLE `branch\_phonenumbers\_relation` ...

### 3.4.12. Donor table

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (foe\_23), Views, Stored Procedures, Functions.
- Table:** donor (selected). The table structure is shown in the main pane.
- Script Editor:** Shows the SQL code for creating the donor table:

```
CREATE TABLE `donor` (
  `Donor_ID` varchar(50) NOT NULL,
  `Donor_Name` varchar(50) NOT NULL,
  `Date_of_Birth` date NOT NULL,
  `Gender` varchar(10) NOT NULL,
  `Nationality` varchar(40) NOT NULL,
  `Eligibility_Status` tinyint(1) DEFAULT '0',
  `Blood_Quantity` int DEFAULT '0',
  `Last_Donated_Date` date DEFAULT NULL,
  `BloodGroup` varchar(3) NOT NULL,
  `Street` varchar(100) NOT NULL,
  `City` varchar(50) NOT NULL,
  `Province` varchar(50) NOT NULL,
  `bloodId` varchar(40) DEFAULT NULL,
  `Age` int DEFAULT NULL,
  PRIMARY KEY (`Donor_ID`),
  KEY `fk_blood1` (`bloodId`),
  CONSTRAINT `fk_blood1` FOREIGN KEY (`bloodId`) REFERENCES `blood` (`Blood_ID`) ON DELETE CASCADE
)
ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```
- Result Grid:** Shows the CREATE TABLE command.
- Output:** Shows the history of recent actions:
  - 268 20:59:13 ALTER TABLE PATIENT ADD Age INT
  - 269 20:59:13 UPDATE PATIENT SET Age = TIMESTAMPDIFF(YEAR, Date\_of\_Birth, NOW())
  - 270 20:59:16 SET SQL\_SAFE\_UPDATES = 1
  - 271 20:59:28 show create table donorWith 0 row(s) affected and 1 row(s) returned.
- Object Info:** Shows the completed query.

### 3.4.13. Donor- Health Problems Table

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (Blood\_Bank\_Management\_System), Views.
- Table:** patient (selected). The table structure is shown in the main pane.
- Script Editor:** Shows the SQL code for creating the donor\_healthproblems\_table:

```
CREATE TABLE `donor_healthproblems_relation` (
  `Donor_ID` varchar(50) NOT NULL,
  `Health_Problems` varchar(250) NOT NULL,
  PRIMARY KEY (`Donor_ID`, `Health_Problems`),
  CONSTRAINT `fk_donor2` FOREIGN KEY (`Donor_ID`) REFERENCES `donor` (`Donor_ID`) ON DELETE CASCADE ON UPDATE CASCADE
)
ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```
- Result Grid:** Shows the CREATE TABLE command.
- Output:** Shows the history of recent actions:
  - 574 20:59:13 show create table blood\_branch\_relation;
  - 575 20:59:13 show create table blood\_donation\_camp;
  - 576 20:59:13 show create table blood\_employee\_relation;
  - 577 20:59:13 show create table blood\_equipment\_relation;
  - 578 20:59:13 show create table blood\_patient\_relation;
  - 579 20:59:13 show create table bloodbank\_hospital\_relation;
  - 580 20:59:13 show create table branch\_availablebloodgroups\_relation;
  - 581 20:59:13 show create table branch\_phonenumbers\_relation;
  - 582 20:59:13 show create table donor;
  - 583 20:59:13 show create table donor\_blooddonationcamp\_relation;
  - 584 20:59:13 show create table donor\_donateddates\_relation;
  - 585 20:59:13 show create table donor\_email\_relation;
  - 586 20:59:13 show create table donor\_healthproblems\_relation;
  - 587 20:59:13 show create table donor\_phonenumbers\_relation;
  - 588 20:59:13 show create table employee;
  - 589 20:59:13 show create table employee\_donor\_relation;
  - 590 20:59:13 show create table employee\_email\_relation;
  - 591 20:59:13 show create table equipment;With 0 row(s) affected and 1 row(s) returned.
- Object Info:** Shows the completed query.

### 3.4.14. Donor- Donated Dates Table

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. In the 'Schemas' tree, the 'patient' schema is expanded, showing its tables and columns. A context menu is open over the 'patient' table, with the 'Create Table' option selected. This has opened a new window titled 'Edit Data for Create Table (VARCHAR)'. The 'Text' tab is active, displaying the SQL code for creating the 'donor\_donateddates\_relation' table:

```
1 CREATE TABLE `donor_donateddates_relation` (
2   `Donor_ID` varchar(50) NOT NULL,
3   `DonatedDate` varchar(50) NOT NULL,
4   PRIMARY KEY (`Donor_ID`, `DonatedDate`),
5   CONSTRAINT `fk_donor` FOREIGN KEY (`Donor_ID`) REFERENCES `donor` (`Donor_ID`) ON DELETE CASCADE ON UPDATE CASCADE
6 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

The 'Result Grid' shows the same SQL code. The status bar at the bottom right indicates 'Data Length: 344 bytes'.

### 3.4.15. Donor- Email Table

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. In the 'Schemas' tree, the 'patient' schema is expanded, showing its tables and columns. A context menu is open over the 'patient' table, with the 'Create Table' option selected. This has opened a new window titled 'Edit Data for Create Table (VARCHAR)'. The 'Text' tab is active, displaying the SQL code for creating the 'donor\_email\_relation' table:

```
1 CREATE TABLE `donor_email_relation` (
2   `Donor_ID` varchar(50) NOT NULL,
3   `Email` varchar(50) NOT NULL,
4   PRIMARY KEY (`Donor_ID`, `Email`),
5   CONSTRAINT `fk_donor` FOREIGN KEY (`Donor_ID`) REFERENCES `donor` (`Donor_ID`) ON DELETE CASCADE ON UPDATE CASCADE
6 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

The 'Result Grid' shows the same SQL code. The status bar at the bottom right indicates 'Data Length: 324 bytes'.

### 3.4.16. Donor- Phone Number Tables

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. In the central query editor, the command to create the 'donor\_phonenumbers\_relation' table is displayed:

```
CREATE TABLE `donor_phonenumbers_relation` (
  `Donor_ID` varchar(50) NOT NULL,
  `Contact_No` varchar(10) NOT NULL,
  PRIMARY KEY (`Donor_ID`, `Contact_No`),
  CONSTRAINT `fk_donor1` FOREIGN KEY (`Donor_ID`) REFERENCES `donor` (`Donor_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

A modal window titled 'Edit Data for Create Table (VARCHAR)' is open, showing the binary representation of the SQL command. The status bar at the bottom right indicates 'Read Only'.

### 3.4.17. Donor- Blood Donation Camp Table

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. In the central query editor, the command to create the 'donor\_blooddonationcamp\_relation' table is displayed:

```
CREATE TABLE `donor_blooddonationcamp_relation` (
  `donor_id` varchar(40) NOT NULL,
  `branch_id` varchar(50) NOT NULL,
  `camp_name` varchar(50) NOT NULL,
  `held_date` date NOT NULL,
  PRIMARY KEY (`donor_id`, `branch_id`, `camp_name`, `held_date`),
  KEY `fk_branch2` (`branch_id`, `camp_name`, `held_date`),
  CONSTRAINT `fk_branch2` FOREIGN KEY (`branch_id`, `camp_name`, `held_date`) REFERENCES `blood_donation_camp` (`Branch_ID`, `Camp_Name`, `Held_Date`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `fk_donor4` FOREIGN KEY (`donor_id`) REFERENCES `donor` (`Donor_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

A modal window titled 'Edit Data for Create Table (VARCHAR)' is open, showing the binary representation of the SQL command. The status bar at the bottom right indicates 'Read Only'.

### 3.4.18. Employee Table

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. In the center, the 'Create Table' dialog is open, displaying the SQL code for creating the 'employee' table. The table structure includes columns for Employee\_ID (PK), Employee\_Name, Gender, Nationality, Date\_of\_Birth, Street, City, and Province. A foreign key constraint 'fk\_bloodbank3' is defined, linking Branch\_ID to Branch\_ID in the blood\_bank table. The code also specifies ENGINE=InnoDB, DEFAULT CHARSET=utf8mb4, and COLLATE=utf8mb4\_0900\_ai\_ci.

```
CREATE TABLE `employee` (
  `Employee_ID` varchar(40) NOT NULL,
  `Employee_Name` varchar(40) NOT NULL,
  `Gender` varchar(10) NOT NULL,
  `Nationality` varchar(40) NOT NULL,
  `Employee_Type` varchar(40) NOT NULL,
  `Date_of_Birth` date NOT NULL,
  `Salary` int DEFAULT '0',
  `Street` varchar(40) NOT NULL,
  `City` varchar(40) NOT NULL,
  `Province` varchar(40) NOT NULL,
  `Supervisor_ID` varchar(40) DEFAULT NULL,
  `Branch_ID` varchar(40) NOT NULL,
  `Age` int DEFAULT NULL,
  PRIMARY KEY (`Employee_ID`),
  UNIQUE KEY `Employee_Name` (`Employee_Name`, `Employee_Type`, `Date_of_Birth`),
  KEY `fk_bloodbank3` (`Branch_ID`),
  CONSTRAINT `fk_bloodbank3` FOREIGN KEY (`Branch_ID`) REFERENCES `blood_bank` (`Branch_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

### 3.4.19. Employee- Phone Numbers

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. In the center, the 'Create Table' dialog is open, displaying the SQL code for creating the 'employee\_phonenumbers\_relation' table. The table structure includes columns for Employee\_ID (PK) and Contact\_No. A foreign key constraint 'fk\_emp1' is defined, linking Employee\_ID to Employee\_ID in the employee table. The code also specifies ENGINE=InnoDB, DEFAULT CHARSET=utf8mb4, and COLLATE=utf8mb4\_0900\_ai\_ci.

```
CREATE TABLE `employee_phonenumbers_relation` (
  `Employee_ID` varchar(40) NOT NULL,
  `Contact_No` varchar(10) NOT NULL,
  PRIMARY KEY (`Employee_ID`, `Contact_No`),
  CONSTRAINT `fk_emp1` FOREIGN KEY (`Employee_ID`) REFERENCES `employee` (`Employee_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

### 3.4.20. Employee- Donor Table Relation

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. In the Navigator pane, the 'Schemas' section is expanded, showing various tables like donor, patient, and employee. The 'Table: patient' section is selected, displaying its columns: Patient\_ID, Patient\_Name, Gender, Nationality, Date\_of\_Birth, Street, City, and Province. A query results grid shows the creation command for the 'employee\_donor\_relation' table:

```
CREATE TABLE `employee_donor_relation` (
  `donor_id` varchar(40) NOT NULL,
  `employee_id` varchar(50) NOT NULL,
  PRIMARY KEY (`donor_id`, `employee_id`),
  KEY `fk_donor3` (`employee_id`),
  CONSTRAINT `fk_donor3` FOREIGN KEY (`donor_id`) REFERENCES `donor` (`Donor_ID`) ON DELETE CASCADE
  ON UPDATE CASCADE,
  CONSTRAINT `fk_empl3` FOREIGN KEY (`employee_id`) REFERENCES `employee` (`Employee_ID`) ON DELETE CASCADE
  ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

A modal window titled 'Edit Data for Create Table (VARCHAR)' is open, showing the same CREATE TABLE statement. The status bar at the bottom right indicates the date and time as 3/31/2024 8:48 PM.

### 3.4.21. Employee- Email Table

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. In the Navigator pane, the 'Schemas' section is expanded, showing various tables like donor, patient, and employee. The 'Table: patient' section is selected, displaying its columns: Patient\_ID, Patient\_Name, Gender, Nationality, Date\_of\_Birth, Street, City, and Province. A query results grid shows the creation command for the 'employee\_email\_relation' table:

```
CREATE TABLE `employee_email_relation` (
  `Employee_ID` varchar(40) NOT NULL,
  `Email` varchar(40) NOT NULL,
  PRIMARY KEY (`Employee_ID`, `Email`),
  CONSTRAINT `fk_empl1` FOREIGN KEY (`Employee_ID`) REFERENCES `employee` (`Employee_ID`) ON DELETE CASCADE
  ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

A modal window titled 'Edit Data for Create Table (VARCHAR)' is open, showing the same CREATE TABLE statement. The status bar at the bottom right indicates the date and time as 3/31/2024 8:48 PM.

### 3.4.22. Equipment Table

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. In the 'Schemas' tree, the 'patient' schema is expanded, showing various tables like 'donor\_phonenumbers\_relation', 'employee', etc. A query window displays the creation of the 'equipment' table:

```
CREATE TABLE `equipment` (
  `Equip_Name` varchar(50) NOT NULL,
  `Cost` int DEFAULT '0',
  `Quantity` int DEFAULT '0',
  `Equip_Description` varchar(250) DEFAULT NULL,
  `Branch_ID` varchar(50) NOT NULL,
  PRIMARY KEY (`Equip_Name`, `Branch_ID`),
  KEY `fk_bloodbank2` (`Branch_ID`),
  CONSTRAINT `fk_bloodbank2` FOREIGN KEY (`Branch_ID`) REFERENCES `blood_bank` (`Branch_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

The 'Result Grid' shows the command being run:

| Table     | Create Table                                 |
|-----------|--|
| equipment | CREATE TABLE `equipment` ( `Equip_Name` v... |

### 3.4.23. Hospital Table

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. In the 'Schemas' tree, the 'patient' schema is expanded, showing various tables like 'donor\_phonenumbers\_relation', 'employee', etc. A query window displays the creation of the 'hospital' table:

```
CREATE TABLE `hospital` (
  `Hospital_ID` varchar(50) NOT NULL,
  `Hospital_Name` varchar(100) NOT NULL,
  `Street` varchar(50) NOT NULL,
  `City` varchar(50) NOT NULL,
  `Province` varchar(50) NOT NULL,
  PRIMARY KEY (`Hospital_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

The 'Result Grid' shows the command being run:

| Table    | Create Table                                   |
|----------|--|
| hospital | CREATE TABLE `hospital` ( `Hospital_ID` var... |

### 3.4.24. Hospital- Phone Numbers Table

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. In the 'Schemas' tree, the 'patient' schema is expanded, and the 'hospital\_phonenumbers\_relation' table is selected. The 'Edit Data for Create Table (VARCHAR)' dialog is open, displaying the SQL code for creating the table:

```
CREATE TABLE `hospital_phonenumbers_relation` (
  `Hospital_ID` varchar(50) NOT NULL,
  `Contact_No` varchar(10) NOT NULL,
  PRIMARY KEY (`Hospital_ID`,`Contact_No`),
  CONSTRAINT `fk_hospital` FOREIGN KEY (`Hospital_ID`) REFERENCES `hospital` (`Hospital_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

The 'Result Grid' shows the CREATE TABLE statement. The system tray at the bottom right indicates the date as 3/31/2024 and the time as 8:49 PM.

### 3.4.25. Patient Table

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. In the 'Schemas' tree, the 'patient' schema is expanded, and the 'patient' table is selected. The 'Edit Data for Create Table (VARCHAR)' dialog is open, displaying the SQL code for creating the table:

```
CREATE TABLE `patient` (
  `Patient_ID` varchar(40) NOT NULL,
  `Patient_Name` varchar(40) NOT NULL,
  `Gender` varchar(10) NOT NULL,
  `Nationality` varchar(40) NOT NULL,
  `Date_of_Birth` date NOT NULL,
  `Street` varchar(40) NOT NULL,
  `City` varchar(40) NOT NULL,
  `Province` varchar(40) NOT NULL,
  `Age` int DEFAULT NULL,
  PRIMARY KEY (`Patient_ID`),
  UNIQUE KEY `Patient_Name` (`Patient_Name`, `Date_of_Birth`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

The 'Result Grid' shows the CREATE TABLE statement. The system tray at the bottom right indicates the date as 3/31/2024 and the time as 8:49 PM.

### 3.4.26. Patient- Phone Numbers Table

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Shows the current database is "Blood Bank Management System" and contains tables like donor, patient, equipment, hospital, etc.
- Table Information:** Shows the "patient" table with columns: Patient\_ID (PK), Patient\_Name, Gender, Nationality, Date\_of\_Birth, Street, City, Province.
- Query Editor:** Displays SQL code for creating the "patient\_phonenumbers\_relation" table. The code is as follows:

```

1 CREATE TABLE `patient_phonenumbers_relation` (
2   `Patient_ID` varchar(40) NOT NULL,
3   `Contact_No` varchar(10) NOT NULL,
4   PRIMARY KEY (`Patient_ID`, `Contact_No`),
5   CONSTRAINT `fk_patient` FOREIGN KEY (`Patient_ID`) REFERENCES `patient` (`Patient_ID`) ON DELETE CASCADE ON UPDATE CASCADE
6 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

```

- Result Grid:** Shows the result of the query, indicating the table was created successfully.
- System Tray:** Shows icons for various applications and system status.

### 3.4.27. Patient- Email Table

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Shows the current database is "Blood Bank Management System" and contains tables like donor, patient, equipment, hospital, etc.
- Table Information:** Shows the "patient" table with columns: Patient\_ID (PK), Patient\_Name, Gender, Nationality, Date\_of\_Birth, Street, City, Province.
- Query Editor:** Displays SQL code for creating the "patient\_email\_relation" table. The code is as follows:

```

1 CREATE TABLE `patient_email_relation` (
2   `Patient_ID` varchar(40) NOT NULL,
3   `Email` varchar(40) NOT NULL,
4   PRIMARY KEY (`Patient_ID`, `Email`),
5   CONSTRAINT `fk_patient` FOREIGN KEY (`Patient_ID`) REFERENCES `patient` (`Patient_ID`) ON DELETE CASCADE ON UPDATE CASCADE
6 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci

```

- Result Grid:** Shows the result of the query, indicating the table was created successfully.
- System Tray:** Shows icons for various applications and system status.

### 3.4.28. Patient- Health Problems Table

The screenshot shows the MySQL Workbench interface with the 'patient' schema selected. A query editor window displays the SQL code for creating the 'patient\_healthproblems\_relation' table. The table has two columns: 'Patient\_ID' (primary key) and 'Health\_Problems' (varchar(250)). It includes a foreign key constraint 'fk\_patient2' referencing the 'Patient\_ID' in the 'patient' table, with cascade operations for update and delete. The engine is set to InnoDB with utf8mb4 charset and collation.

```
CREATE TABLE `patient_healthproblems_relation` (
  `Patient_ID` varchar(40) NOT NULL,
  `Health_Problems` varchar(250) NOT NULL,
  PRIMARY KEY (`Patient_ID`),
  CONSTRAINT `fk_patient2` FOREIGN KEY (`Patient_ID`) REFERENCES `patient` (`Patient_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

### 3.4.29. Payment Table

The screenshot shows the MySQL Workbench interface with the 'patient' schema selected. A query editor window displays the SQL code for creating the 'payment' table. The table has four columns: 'Payment\_ID' (primary key), 'Amount' (decimal(3,1)), 'Paid\_Date' (date), and 'Patient\_ID' (foreign key). The foreign key 'fk\_patient3' references the 'Patient\_ID' in the 'patient' table, with cascade operations for update and delete. The engine is set to InnoDB with utf8mb4 charset and collation.

```
CREATE TABLE `payment` (
  `Payment_ID` varchar(50) NOT NULL,
  `Amount` decimal(3,1) NOT NULL,
  `Paid_Date` date NOT NULL,
  `Patient_ID` varchar(50) NOT NULL,
  PRIMARY KEY (`Payment_ID`),
  KEY `fk_patient3` (`Patient_ID`),
  CONSTRAINT `fk_patient3` FOREIGN KEY (`Patient_ID`) REFERENCES `patient` (`Patient_ID`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

## 3.5. Updating & Deleting data of tables

### 3.5.1. Blood table

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. In the center, the 'patient' table is displayed in a result grid. The columns shown are Blood\_ID, Blood\_Group, Expiry\_Date, Blood\_Quantity\_ml, and Blood\_Description. The data includes rows for various blood units with details like O+, A+, B+, AB+, O-, and B-. The bottom status bar shows the date as 3/31/2024 and the time as 10:32 PM.

| Blood_ID | Blood_Group | Expiry_Date | Blood_Quantity_ml | Blood_Description |
|----------|-------------|-------------|-------------------|-------------------|
| B33467   | O+          | 2024-06-25  | 390               | stored            |
| B33469   | A+          | 2024-07-11  | 500               | stored            |
| B33470   | B+          | 2024-06-15  | 450               | stored            |
| B33486   | A-          | 2024-05-20  | 400               | stored            |
| B33492   | AB+         | 2024-06-02  | 385               | stored            |
| B33493   | O+          | 2024-06-25  | 470               | stored            |
| B33500   | O-          | 2024-08-07  | 430               | stored            |
| NULL     | NULL        | NULL        | NULL              | NULL              |

### 3.5.2. Blood - Branch Relation Table

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. In the center, the 'blood\_branch\_relation' table is displayed in a result grid. The columns shown are blood\_id and branch\_id. The data includes rows for various blood units associated with different branches, such as Br009, Br011, Br014, Br016, Br017, Br021, Br024, and Br037. The bottom status bar shows the date as 3/31/2024 and the time as 10:34 PM.

| blood_id | branch_id |
|----------|-----------|
| B33486   | Br009     |
| B33467   | Br011     |
| B33469   | Br014     |
| B33492   | Br016     |
| B33486   | Br017     |
| B33469   | Br021     |
| B33469   | Br024     |
| B33500   | Br024     |
| B33470   | Br037     |
| NULL     | NULL      |

### 3.5.3. Blood - Blood Donation Camps Table

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas patient patient

Limit to 1000 rows

648  
649 • select \* from blood\_donation\_camp;  
650 • update blood\_donation\_camp set Street = 'Street9' where Branch\_ID = 'B33486' and Camp\_Name = 'Camp1' and Held\_Date = '2024-01-30';  
651 • update blood\_donation\_camp set Street = 'Street10' where Branch\_ID = 'B33470' and Camp\_Name = 'Camp4' and Held\_Date = '2024-01-22';  
652 • delete from blood\_donation\_camp where Branch\_ID = 'B009' and Camp\_Name = 'Camp5' and Held\_Date = '2024-02-17';  
653 • select \* from blood\_donation\_camps;

654  
655  
656  
657  
658  
659  
660

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

| Branch_ID | Camp_Name | Held_Date  | Street  | City         | Province      |
|-----------|-----------|------------|---------|--------------|---------------|
| Br011     | Camp1     | 2024-01-30 | Street1 | Anuradhapura | North Central |
| Br014     | Camp4     | 2024-01-22 | Street4 | Kurunegala   | North Western |
| Br016     | Camp8     | 2024-01-02 | Street8 | Colombo      | Western       |
| Br021     | Camp6     | 2024-03-12 | Street6 | Peraudiya    | Central       |
| Br024     | Camp3     | 2024-01-13 | Street3 | Jaffna       | Northern      |
| Br037     | Camp7     | 2024-02-20 | Street7 | Kandy        | Central       |
| NULL      | NULL      | NULL       | NULL    | NULL         | NULL          |

blood\_donation\_camp\_44 blood\_donation\_camp\_45

Object Info Session Output

Query Completed

31°C Mostly clear

Search

10:36 PM 3/31/2024

### 3.5.4. Blood- Employee Relation Table

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas patient patient

Limit to 1000 rows

660  
661 • select \* from blood\_employee\_relation;  
662 • update blood\_employee\_relation set blood\_id = 'B33470' where blood\_id = 'B33486' and employee\_id = 'E045';  
663 • update blood\_employee\_relation set blood\_id = 'B33486' where blood\_id = 'B33470' and employee\_id = 'E053';  
664 • delete from blood\_employee\_relation where blood\_id = 'B33467' and employee\_id = 'E034';  
665 • select \* from blood\_employee\_relation;

666  
667  
668  
669  
670  
671  
672

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

| blood_id | employee_id |
|----------|-------------|
| B33469   | E045        |
| B33470   | E045        |
| B33486   | E053        |
| B33500   | E058        |
| B33493   | E066        |
| B33467   | E072        |
| B33492   | E072        |
| B33500   | E072        |
| NULL     | NULL        |

blood\_employee\_relation\_46 blood\_employee\_relation\_47

Object Info Session Output

Query Completed

31°C Mostly clear

Search

10:37 PM 3/31/2024

### 3.5.5. Blood- Equipment Relation Table

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

SCHEMAS

Table: patient

Columns:

| Patient_ID     | Patient_Name | Gender      | Nationality | Date_of_Birth | Street      | City        | Province    |
|----------------|--------------|-------------|-------------|---------------|-------------|-------------|-------------|
| varchar(40) PK | varchar(40)  | varchar(10) | varchar(40) | date          | varchar(40) | varchar(40) | varchar(40) |

Table: blood\_equipment\_relation

Columns:

| blood_id | Eqip_Name                    | Branch_ID |
|----------|------------------------------|-----------|
| B33469   | Blood bag tube sealer        | Br014     |
| B33469   | Blood donor couch            | Br024     |
| B33469   | Blood mixer roller           | Br009     |
| B33470   | Blood cell separator machine | Br011     |
| B33470   | Blood filters                | Br021     |
| B33486   | Blood circulation machine    | Br016     |
| B33486   | Blood collection monitor     | Br011     |
| B33492   | Blood cell separator machine | Br011     |
| B33493   | Blood collection monitor     | Br011     |
| B33500   | Blood donor couch            | Br024     |
| B33500   | Blood mixer roller           | Br009     |
| NULL     | NULL                         | NULL      |

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

Object Info Session

Query Completed

31°C Mostly clear

Search

10:37 PM 3/31/2024

### 3.5.6. Blood- Patient Table

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

SCHEMAS

Table: patient

Columns:

| Patient_ID     | Patient_Name | Gender      | Nationality | Date_of_Birth | Street      | City        | Province    |
|----------------|--------------|-------------|-------------|---------------|-------------|-------------|-------------|
| varchar(40) PK | varchar(40)  | varchar(10) | varchar(40) | date          | varchar(40) | varchar(40) | varchar(40) |

Table: blood\_patient\_relation

Columns:

| blood_id | patient_id |
|----------|------------|
| B33486   | P23488     |
| B33470   | P23489     |
| B33493   | P23490     |
| B33500   | P23491     |
| NULL     | NULL       |

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

Object Info Session

Query Completed

31°C Mostly clear

Search

10:37 PM 3/31/2024

### 3.5.7. Blood- Test Samples Relation Table

The screenshot shows the MySQL Workbench interface with a query editor and a results grid.

**Query Editor:**

```

624 • select * from blood_bloodtestsamples_relation;
625 • update blood_bloodtestsamples_relation set Test_Done_For_Blood_Samples = 'Basic Metabolic Panel' where Blood_ID = 'B33469' and Test_Done_For_Blood_Samples = 'Basic Metal';
626 • update blood_bloodtestsamples_relation set Test_Done_For_Blood_Samples = 'Lipid Panel' where Blood_ID = 'B33469' and Test_Done_For_Blood_Samples = 'Complete Blood Count';
627 • delete from blood_bloodtestsamples_relation where Blood_ID = 'Br003';
628 • select * from blood_bloodtestsamples_relation;
629
630
631
632
633
634
635
636

```

**Results Grid:**

| Blood_ID | Test_Done_For_Blood_Samples |
|----------|-----------------------------|
| B33467   | Lipid Panel                 |
| B33469   | Basic Metabolic Panel       |
| B33469   | Lipid Panel                 |
| B33470   | Basic Metabolic Panel       |
| B33486   | Complete Blood Count        |
| B33486   | Lipid Panel                 |
| B33493   | Arterial Blood Gas          |
| B33493   | Basic Metabolic Panel       |
| B33493   | Complete Blood Count        |
| B33500   | Arterial Blood Gas          |
| NULL     | NULL                        |

### 3.5.8. Blood Bank Table

The screenshot shows the MySQL Workbench interface with a query editor and a results grid.

**Query Editor:**

```

612 • select * from blood_bank;
613 • update blood_bank set Available_Blood_Quantity = 5400 where Branch_ID = 'Br009';
614 • update blood_bank set Available_Blood_Quantity = 7508 where Branch_ID = 'Br011';
615 • delete from blood_bank where Branch_ID = 'Br003';
616 • select * from blood_bank;
617
618 -- ----- DONE -----
619
620
621
622
623
624

```

**Results Grid:**

| Branch_ID | Branch_Name   | Available_Blood_Quantity | Street  | City         | Province      |
|-----------|---------------|--------------------------|---------|--------------|---------------|
| Br009     | Diyathalawa   | 5400                     | Street5 | Badulla      | Uva           |
| Br011     | Anuradhapura  | 7508                     | Street1 | Anuradhapura | North Central |
| Br014     | Dambadeniya   | 4830                     | Street4 | Kurunegala   | North Western |
| Br016     | Army Hospital | 9850                     | Street8 | Colombo      | Western       |
| Br021     | Gampola       | 6890                     | Street6 | Peradeniya   | Central       |
| Br024     | Kilinochchi   | 3400                     | Street3 | Jaffna       | Northern      |
| Br037     | Dambulla      | 0                        | Street7 | Kandy        | Central       |
| NULL      | NULL          | NULL                     | NULL    | NULL         | NULL          |

### 3.5.9. Blood Bank- Hospital Relation Table

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** The schema browser on the left lists tables such as donor\_phonenumbers\_relation, employee, employee\_donor\_relation, employee\_email\_relation, employee\_phonenumbers\_re, equipment, hospital, hospital\_phonenumbers\_relz, patient, patient\_email\_relation, patient\_healthproblems\_rela, patient\_phonenumbers\_relat, and payment.
- Query Editor:** The main window displays a series of SQL commands for updating the bloodbank\_hospital\_relation table. The commands include selecting all rows, updating rows where hospital\_id = 'H008' and branch\_id = 'Br009', updating rows where hospital\_id = 'H022' and branch\_id = 'Br011', deleting rows where hospital\_id = 'H011' and branch\_id = 'Br009', and finally selecting all rows again.
- Result Grid:** Below the query editor, the result grid shows the updated data in the bloodbank\_hospital\_relation table. The columns are hospital\_id and branch\_id. The data includes rows for H008, H019, H022, H011, H011, H019, H016, H015, H021, H010, H024, H016, and H016, with some entries being null.
- Information:** On the left, the 'Information' panel shows the structure of the patient table, including columns like Patient\_ID, Patient\_Name, Gender, Nationality, Date\_of\_Birth, Street, City, and Province.
- Object Info:** The bottom left shows tabs for Object Info and Session.
- System Bar:** The bottom right shows system status icons for temperature (31°C), battery, signal strength, and date/time (3/31/2024, 10:38 PM).

### 3.5.10. Branch- Available Blood Groups Table

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** The schema browser on the left lists tables such as donor\_phonenumbers\_relation, employee, employee\_donor\_relation, employee\_email\_relation, employee\_phonenumbers\_re, equipment, hospital, hospital\_phonenumbers\_relz, patient, patient\_email\_relation, patient\_healthproblems\_rela, patient\_phonenumbers\_relat, and payment.
- Query Editor:** The main window displays a series of SQL commands for updating the branch\_availablebloodgroups\_relation table. The commands include selecting all rows, updating rows where Available\_Blood\_Groups = 'B-' and Available\_Blood\_Quantity = '2505', updating rows where Available\_Blood\_Groups = 'O-' and Available\_Blood\_Quantity = '5005', deleting rows where Available\_Blood\_Groups = 'O-', and finally selecting all rows again.
- Result Grid:** Below the query editor, the result grid shows the updated data in the branch\_availablebloodgroups\_relation table. The columns are Branch\_ID, Available\_Blood\_Groups, and Available\_Blood\_Quantity. The data includes rows for Br011, Br011, Br014, Br014, Br014, Br016, Br016, Br021, Br021, Br024, and Br024, with values like B-, O-, A-, O+, A+, B-, O-, B+, O+, and A+ respectively.
- Information:** On the left, the 'Information' panel shows the structure of the patient table, including columns like Patient\_ID, Patient\_Name, Gender, Nationality, Date\_of\_Birth, Street, City, and Province.
- Object Info:** The bottom left shows tabs for Object Info and Session.
- System Bar:** The bottom right shows system status icons for temperature (31°C), battery, signal strength, and date/time (3/31/2024, 10:38 PM).

### 3.5.11. Branch- Phone Numbers Table

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas patient patient

```

220
221 • select * from branch_phonenumbers_relation;
222 • update branch_phonenumbers_relation set Contact_No = '0112456287' where Branch_ID = 'Br011' and Contact_No = '0112456288';
223 • update branch_phonenumbers_relation set Contact_No = '0112856949' where Branch_ID = 'Br011' and Contact_No = '0112856947';
224 • delete from branch_phonenumbers_relation where Branch_ID = 'Br009' and Contact_No = '0112249637';
225 • select * from branch_phonenumbers_relation;
226
227
228
229
230
231
232

```

Result Grid | Filter Rows: | Edit | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor | Field Types | Query Stats |

Table: patient

Columns:

| Patient_ID | Patient_Name | Gender | Nationality | Date_of_Birth | Street | City | Province |
|------------|--------------|--------|-------------|---------------|--------|------|----------|
| Br011      | 0112456287   |        |             |               |        |      |          |
| Br011      | 0112856949   |        |             |               |        |      |          |
| Br014      | 0112796458   |        |             |               |        |      |          |
| Br016      | 0112266657   |        |             |               |        |      |          |
| Br021      | 0112111597   |        |             |               |        |      |          |
| Br024      | 0112954126   |        |             |               |        |      |          |
| Br037      | 0112644799   |        |             |               |        |      |          |
| NULL       | NULL         |        |             |               |        |      |          |

branch\_phonenumbers\_relation... branch\_phonenumbers\_relation... | Apply | Revert

Object Info Session Output

Query Completed

31°C Mostly clear

Search | Windows Taskbar | System Tray | 10:38 PM 3/31/2024

### 3.5.12. Donor table

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas patient patient

```

733 • select * from donor;
734 • update donor set Available_Blood_Quantity = 5400 where Branch_ID = 'Br009';
735 • update donor set Available_Blood_Quantity = 7508 where Branch_ID = 'Br011';
736 • delete from donor where Branch_ID = 'Br003';
737 • select * from donor;
738
739
740
741
742
743
744
745

```

Result Grid | Filter Rows: | Edit | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor | Field Types | Query Stats |

Table: patient

Columns:

| Patient_ID | Patient_Name | Gender | Nationality | Eligibility_Status | Blood_Quantity | Last_Donated_Date | Blood_Group | Street                  | City         | Province      | Blood_ID | Age  | Gender |      |
|------------|--------------|--------|-------------|--------------------|----------------|-------------------|-------------|-------------------------|--------------|---------------|----------|------|--------|------|
| D651354    | Dineth       | Male   | Sinhalese   | 1                  | 390            | 2022-03-25        | O+          | Hilogama                | Kurunegala   | North Western | B33467   | 22   | Male   |      |
| D651355    | Tharushi     | Female | Sinhalese   | 1                  | 500            | 2023-10-20        | A+          | Udagama                 | Matale       | Central       | B33469   | 23   | Male   |      |
| D651357    | Kavya        | Female | Tamil       | 1                  | 450            | 2023-12-04        | B+          | Thirunagar North        | Kilinochchi  | Northern      | B33470   | 21   | Male   |      |
| D651360    | Nimaya       | Female | Sinhalese   | 1                  | 400            | 2022-01-30        | A-          | Hurigasewwa             | Anuradhapura | North Central | B33486   | 23   | Male   |      |
| D651362    | Yugan        | Male   | Tamil       | 1                  | 385            | 2020-10-06        | O-          | Barathypuram            | Kilinochchi  | Northern      | B33492   | 23   | Male   |      |
| D651366    | Aravind      | Male   | Tamil       | 1                  | 470            | 2022-05-27        | O+          | Parantan-Mullaibuvu Hwy | Tanniyuttu   | Northern      | B33493   | 22   | Male   |      |
| D651369    | Kavindu      | Male   | Sinhalese   | 1                  | 430            | 2022-04-05        | O-          | Rupaha                  | Nuwara Eliya | Central       | B33500   | 24   | Male   |      |
| NULL       | NULL         | NULL   | NULL        | NULL               | NULL           | NULL              | NULL        | NULL                    | NULL         | NULL          | NULL     | NULL | NULL   | NULL |

donor 63 | Apply | Revert

Object Info Session Output

Query interrupted

31°C Mostly clear

Search | Windows Taskbar | System Tray | 10:40 PM 3/31/2024

### 3.5.13. Donor- Health Problems Table

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** A tree view showing various database objects like donor\_phonenumbers\_relation, employee, etc.
- Query Editor:** Contains several SQL statements related to the donor\_healthproblems\_table:
  - 787 • select \* from donor\_healthproblems\_relation;
  - 788 • update donor\_healthproblems\_relation set Health\_Problems = 'Migraines' where Donor\_ID = 'D651357' and Health\_Problems = 'Acne';
  - 789 • update donor\_healthproblems\_relation set Health\_Problems = 'Acne' where Donor\_ID = 'D651360' and Health\_Problems = 'Migraines';
  - 790 • delete from donor\_healthproblems\_relation where Donor\_ID = 'D651355' and Health\_Problems = 'none';
  - 791 • select \* from donor\_healthproblems\_relation;
- Result Grid:** Displays the data from the donor\_healthproblems\_table:

| Donor_ID | Health_Problems |
|----------|-----------------|
| D651354  | Mild Anemia     |
| D651357  | Migraines       |
| D651360  | Acne            |
| D651362  | none            |
| D651366  | Hypertension    |
| D651369  | none            |
| NULL     | NULL            |
- Information:** Shows the definition of the patient table:

```
Table: patient
Columns:
Patient_ID      varchar(40) PK
Patient_Name    varchar(40)
Gender          varchar(10)
Nationality     varchar(40)
Date_of_Birth   date
Street          varchar(40)
City            varchar(40)
Province        varchar(40)
```
- Object Info:** Shows session information and system status.

### 3.5.14. Donor- Donated Dates Table

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** A tree view showing various database objects like donor\_phonenumbers\_relation, employee, etc.
- Query Editor:** Contains several SQL statements related to the donor\_donateddates\_table:
  - 760
  - 761
  - 762
  - 763 • select \* from donor\_donateddates\_relation;
  - 764 • update donor\_donateddates\_relation set DonatedDate = '2024-01-22' where Donor\_ID = 'D651357' and DonatedDate = '2024-01-13';
  - 765 • update donor\_donateddates\_relation set DonatedDate = '2024-01-13' where Donor\_ID = 'D651360' and DonatedDate = '2024-01-22';
  - 766 • delete from donor\_donateddates\_relation where Donor\_ID = 'D651355' and DonatedDate = '2024-02-10';
  - 767 • select \* from donor\_donateddates\_relation;
- Result Grid:** Displays the data from the donor\_donateddates\_table:

| Donor_ID | DonatedDate |
|----------|-------------|
| D651354  | 2024-01-30  |
| D651357  | 2024-01-22  |
| D651360  | 2024-01-13  |
| D651362  | 2024-02-17  |
| D651366  | 2024-03-12  |
| D651369  | 2024-01-02  |
| NULL     | NULL        |
- Information:** Shows the definition of the patient table:

```
Table: patient
Columns:
Patient_ID      varchar(40) PK
Patient_Name    varchar(40)
Gender          varchar(10)
Nationality     varchar(40)
Date_of_Birth   date
Street          varchar(40)
City            varchar(40)
Province        varchar(40)
```
- Object Info:** Shows session information and system status.

### 3.5.15. Donor- Email Table

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. The 'donor\_email\_relation' table is currently open, displaying a grid of data. The columns are 'Donor\_ID' and 'Email'. The data includes several rows of email addresses corresponding to different donor IDs.

| Donor_ID | Email                 |
|----------|-----------------------|
| D651354  | dineth2001@gmail.com  |
| D651357  | kavyamm2002@gmail.com |
| D651360  | nimaya2008@gmail.com  |
| D651362  | yugan2000@gmail.com   |
| D651366  | aravind2001@gmail.com |
| D651369  | kavindu2002@gmail.com |
| NULL     | NULL                  |

### 3.5.16. Donor- Phone Number Tables

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. The 'donor\_phonenumbers\_relation' table is currently open, displaying a grid of data. The columns are 'Donor\_ID' and 'Contact\_No'. The data includes several rows of contact numbers corresponding to different donor IDs.

| Donor_ID | Contact_No |
|----------|------------|
| D651354  | 0771256948 |
| D651357  | 0777458620 |
| D651360  | 0784596327 |
| D651362  | 0704569881 |
| D651366  | 0712582588 |
| D651369  | 0712546999 |
| NULL     | NULL       |

### 3.5.17. Donor- Blood Donation Camp Table

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

```

751 • select * from donor_blooddonationcamp_relation;
752 • update donor_blooddonationcamp_relation set donor_id = 'D651357' where donor_id = 'D651362' and camp_name = 'Camp6' and branch_id = 'Br021' and held_date = '2024-03-12';
753 • update donor_blooddonationcamp_relation set donor_id = 'D651362' where donor_id = 'D651357' and camp_name = 'Camp3' and branch_id = 'Br024' and held_date = '2024-01-13';
754 • delete from donor_blooddonationcamp_relation where held_date = '2024-01-22' and donor_id = 'D651360' and camp_name = 'Camp4' and branch_id = 'Br014';
755 • select * from donor_blooddonationcamp_relation;
756
757
758
759
760
761
762
763 • select * from donor_donateddates_relation;
    
```

**Result Grid:**

| donor_id | branch_id | camp_name | held_date  |
|----------|-----------|-----------|------------|
| D651354  | Br011     | Camp1     | 2024-01-30 |
| D651369  | Br016     | Camp8     | 2024-01-02 |
| D651357  | Br021     | Camp6     | 2024-03-12 |
| D651362  | Br024     | Camp3     | 2024-01-13 |
| D651366  | Br037     | Camp7     | 2024-02-20 |

### 3.5.18. Employee Table

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

```

812
813
814 • select * from employee;
815 • update employee set Employee_Name = 'Kamalan' where Employee_ID = 'E034';
816 • update employee set Employee_Name = 'Keerthi Suresh' where Employee_ID = 'E045';
817 • delete from employee where Employee_ID = 'D651354';
818 • select * from employees;
819
820
821
822
823
824
    
```

**Result Grid:**

| Employee_ID | Employee_Name  | Gender | Nationality | Employee_Type                   | Date_of_Birth | Salary | Street   | City         | Province      | Supervisor_ID | Branch_ID | Age |
|-------------|----------------|--------|-------------|---------------------------------|---------------|--------|----------|--------------|---------------|---------------|-----------|-----|
| E008        | Chamila        | Female | Sinhalese   | Customer Service Representative | 1990-03-21    | 40000  | Street33 | Badulla      | Uva           | 542           | Br009     | 39  |
| E034        | Kamalan        | Male   | Sinhalese   | Administrative Staff            | 1990-04-29    | 50000  | Street63 | Anuradhapura | North Central | 524           | Br011     | 33  |
| E045        | Keerthi Suresh | Male   | Sinhalese   | Nurse                           | 1983-09-21    | 80000  | Street60 | Kurunegala   | North Western | 541           | Br014     | 40  |
| E053        | Kanthi         | Female | Tamil       | Information Technology Support  | 1991-10-11    | 100000 | Street50 | Jaffna       | Northern      | 534           | Br024     | 32  |
| E058        | Sunil          | Male   | Sinhalese   | Nurse                           | 1985-07-23    | 90000  | Street74 | Colombo      | Western       | 556           | Br016     | 38  |
| E066        | Ranjith        | Male   | Sinhalese   | Research Scientist              | 1968-10-27    | 150000 | Street32 | Kandy        | Central       | 552           | Br037     | 55  |
| E072        | Pradeep        | Male   | Sinhalese   | Administrative Staff            | 1984-09-21    | 60000  | Street54 | Peradeniya   | Central       | 547           | Br021     | 39  |
| E073        | Shyama         | Female | Sinhalese   | Nurse                           | 1982-12-21    | 80000  | Street55 | Peradeniya   | Central       | 548           | Br021     | 41  |

### 3.5.19. Employee- Phone Numbers

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** The "patient" schema is selected.
- Code Editor:** Contains SQL queries for updating the "employee\_phonenumbers\_relation" table. The queries include selecting all rows, updating rows where Employee\_ID = 'E034' and Contact\_No = '0776581414', updating rows where Employee\_ID = 'E045' and Contact\_No = '0774573211', deleting a row where Employee\_ID = 'E008' and Contact\_No = '0784545552', and selecting all rows again.
- Result Grid:** Displays the contents of the "Employee\_ID" and "Contact\_No" columns from the "employee\_phonenumbers\_relation" table. The data includes rows for E034, E045, E053, E058, E066, E072, E073, and a NULL entry.
- Information:** Shows the structure of the "patient" table, which has columns: Patient\_ID (PK), Patient\_Name, Gender, Nationality, Date\_of\_Birth, Street, City, and Province.
- Object Info:** Shows the status of the "employee\_phonenumbers\_relation" object.
- Output:** Shows the status of the "employee\_phonenumbers\_relation" object.
- System Bar:** Includes weather information (31°C), system icons, and system status (10:42 PM, 3/31/2024).

### 3.5.20. Employee- Donor Table Relation

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** The "patient" schema is selected.
- Code Editor:** Contains SQL queries for updating the "employee\_donor\_relation" table. The queries include selecting all rows, updating rows where employee\_id = 'E066' and donor\_id = 'D651357', updating rows where employee\_id = 'E045' and donor\_id = 'D651366', deleting a row where donor\_id = 'D651360' and employee\_id = 'E008', and selecting all rows again.
- Result Grid:** Displays the contents of the "donor\_id" and "employee\_id" columns from the "employee\_donor\_relation" table. The data includes rows for D651354, D651366, D651369, D651357, D651362, and a NULL entry.
- Information:** Shows the structure of the "patient" table, which has columns: Patient\_ID (PK), Patient\_Name, Gender, Nationality, Date\_of\_Birth, Street, City, and Province.
- Object Info:** Shows the status of the "employee\_donor\_relation" object.
- Output:** Shows the status of the "employee\_donor\_relation" object.
- System Bar:** Includes weather information (31°C), system icons, and system status (10:42 PM, 3/31/2024).

### 3.5.21. Employee- Email Table

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Navigator pane shows tables like donor\_phonenumbers\_relation, employee, employee\_donor\_relation, etc.
- Query Editor:** Contains several SQL statements for updating the employee\_email\_relation table.
- Result Grid:** Shows the results of a query on the patient table, listing columns Employee\_ID and Contact\_No with data for E034 through E073.
- Information:** Shows the definition of the patient table.
- Object Info:** Shows session information.
- Output:** Shows the status "Query Completed".
- System Bar:** Includes weather (31°C), search, and system icons.

### 3.5.22. Equipment Table

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Navigator pane shows tables like donor\_phonenumbers\_relation, employee, equipment, etc.
- Query Editor:** Contains several SQL statements for updating the equipment table.
- Result Grid:** Shows the results of a query on the equipment table, listing columns Eqip\_Name, Cost, Quantity, Equip\_Description, and Branch\_ID with data for various medical equipment items.
- Information:** Shows the definition of the patient table.
- Object Info:** Shows session information.
- Output:** Shows the status "Query Completed".
- System Bar:** Includes weather (31°C), search, and system icons.

### 3.5.23. Hospital Table

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

SCHEMAS

donor\_phonenumbers\_relation employee employee\_donor\_relation employee\_email\_relation employee\_phonenumbers\_re equipment hospital hospital\_phonenumbers\_rel patient patient\_email\_relation patient\_healthproblems\_rel patient\_phonenumbers\_rel payment

Blood Bank Management Syst... donor patient patient

872  
873 • select \* from hospital;  
874 • update hospital set Street = 'StreetH10/2' where Hospital\_ID = 'H010';  
875 • update hospital set Street = 'StreetH8/2' where Hospital\_ID = 'H011';  
876 • delete from hospital where Hospital\_ID = 'H008';  
877 • select \* from hospital;

878  
879  
880  
881  
882  
883  
884

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content: Result Grid Form Editor Field Types Query Stats

Table: patient

Columns:

| Patient_ID | Patient_Name                 | Street   | City        | Province      |
|------------|------------------------------|----------|-------------|---------------|
| H008       | General Hospital             | StreetH1 | Kandy       | Central       |
| H010       | Co-operative Hospital        | StreetH2 | Matale      | Central       |
| H011       | Kantala Base Hospital        | StreetH3 | Trincomalee | Eastern       |
| H015       | Kurunegala Hospital          | StreetH4 | Kurunegala  | North Western |
| H016       | Chilaw General Hospital      | StreetH5 | Puttalam    | North Western |
| H019       | Rambukkana District Hospital | StreetH6 | Ratnapura   | Sabaragamuwa  |
| H022       | Karapitiya teaching Hospital | StreetH7 | Galle       | Southern      |
| NULL       | NULL                         | NULL     | NULL        | NULL          |

Object Info Session Output

Query interrupted

31°C Mostly clear

Search

10:43 PM 3/31/2024

### 3.5.24. Hospital- Phone Numbers Table

MySQL Workbench Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

SCHEMAS

donor\_phonenumbers\_relation employee employee\_donor\_relation employee\_email\_relation employee\_phonenumbers\_re equipment hospital hospital\_phonenumbers\_rel patient patient\_email\_relation patient\_healthproblems\_rel patient\_phonenumbers\_rel payment

Blood Bank Management Syst... donor patient patient

884  
885 • select \* from hospital\_phonenumbers\_relation;  
886 • update hospital\_phonenumbers\_relation set Contact\_No = '0112495711' where Hospital\_ID = 'H011' and Contact\_No = '0112887955';  
887 • update hospital\_phonenumbers\_relation set Contact\_No = '0112887955' where Hospital\_ID = 'H015' and Contact\_No = '0112495711';  
888 • delete from hospital\_phonenumbers\_relation where Hospital\_ID = 'H010' and Hospital\_ID = '0112334759';  
889 • select \* from hospital\_phonenumbers\_relations;

890  
891  
892  
893  
894  
895  
896

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content: Result Grid Form Editor Field Types Query Stats

Table: patient

Columns:

| Hospital_ID | Contact_No |
|-------------|------------|
| H008        | 0112345676 |
| H010        | 0112334759 |
| H011        | 0112495711 |
| H015        | 0112887955 |
| H016        | 0112151484 |
| H019        | 0112622522 |
| H022        | 0112860960 |
| NULL        | NULL       |

Object Info Session Output

Query Completed

31°C Mostly clear

Search

10:49 PM 3/31/2024

### 3.5.25. Patient Table

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. The 'patient' table is currently open, displaying the following data:

| Patient_ID | Patient_Name  | Gender | Nationality | Date_of_Birth | Street   | City           | Province | Age |
|------------|---------------|--------|-------------|---------------|----------|----------------|----------|-----|
| P23488     | Kavindu Manuj | Male   | Sinhalese   | 1995-11-09    | Street81 | Bandarawela    | Uva      | 28  |
| P23489     | Ranima        | Female | Sinhalese   | 1976-03-21    | Street57 | Hatton         | Central  | 48  |
| P23490     | Pramtha       | Male   | Tamil       | 1970-12-24    | Street32 | Chavakachcheri | Northern | 53  |
| P23491     | Samitha       | Female | Sinhalese   | 2001-01-30    | Street84 | Peliyagoda     | Western  | 36  |

The SQL editor at the top contains the following code:

```
896  
897  
898 • select * from patient;  
899 • update patient set Patient_Name = 'Kavindu Manuj' where Patient_ID = 'P23488';  
900 • update patient set Patient_Name = 'Ranima' where Patient_ID = 'P23489';  
901 • delete from patient where Patient_ID = 'P23487';  
902 • select * from patient;  
903  
904  
905  
906  
907  
908
```

### 3.5.26. Patient- Phone Numbers Table

The screenshot shows the MySQL Workbench interface with the 'Blood Bank Management System' database selected. The 'patient\_phonenumbers\_relation' table is currently open, displaying the following data:

| Patient_ID | Contact_No |
|------------|------------|
| P23489     | 0784596327 |
| P23490     | 0777458620 |
| P23491     | 0701477899 |

The SQL editor at the top contains the following code:

```
932  
933  
934 • select * from patient_phonenumbers_relation;  
935 • update patient_phonenumbers_relation set Contact_No = '0777458620' where Patient_ID = 'P23490' and Contact_No = '0774115112';  
936 • update patient_phonenumbers_relation set Contact_No = '0784596327' where Patient_ID = 'P23489' and Contact_No = '0772582588';  
937 • delete from patient_phonenumbers_relation where Patient_ID = 'P23488' and Contact_No = '0714544841';  
938 • select * from patient_phonenumbers_relation;  
939  
940  
941  
942  
943  
944
```

### 3.5.27. Patient- Email Table

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Shows the schema tree for the "Blood Bank Management System" database, including tables like donor\_phonenumbers\_relation, employee, employee\_donor\_relation, etc.
- Query Editor:** Displays the following SQL code:

```
908 • select * from patient_email_relation;
910 • update patient_email_relation set Email = 'rani011@gmail.com' where Patient_ID = 'P23489' and Email = 'rani@gmail.com';
911 • update patient_email_relation set Email = 'pramitha005@gmail.com' where Patient_ID = 'P23490' and Email = 'pramitha@gmail.com';
912 • delete from patient_email_relation where Patient_ID = 'P23488' and Email = 'kavindu@gmail.com';
913 • select * from patient_email_relation;
```
- Result Grid:** Shows the results of the last query, displaying three rows of data:

| Patient_ID | Email                 |
|------------|-----------------------|
| P23489     | rani011@gmail.com     |
| P23490     | pramitha005@gmail.com |
| P23491     | samitha@gmail.com     |
- Information Panel:** Shows the definition of the "patient" table:

```
Table: patient
Columns:
Patient_ID varchar(40) PK
Patient_Name varchar(40)
Gender varchar(10)
Nationality varchar(40)
Date_of_Birth date
Street varchar(40)
City varchar(40)
Province varchar(40)
```
- Status Bar:** Shows the weather as 31°C Mostly clear, system icons, and the date/time as 10:50 PM 3/31/2024.

### 3.5.28. Patient- Health Problems Table

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Shows the schema tree for the "Blood Bank Management System" database, including tables like donor\_phonenumbers\_relation, employee, employee\_donor\_relation, etc.
- Query Editor:** Displays the following SQL code:

```
920
921 • select * from patient_healthproblems_relation;
923 • update patient_healthproblems_relation set Health_Problems = 'Surgery' where Patient_ID = 'P23489' and Health_Problems = 'Bone marrow failure';
924 • update patient_healthproblems_relation set Health_Problems = 'Bone marrow failure' where Patient_ID = 'P23488' and Health_Problems = 'Hemorrhage';
925 • delete from patient_healthproblems_relation where Patient_ID = 'P23488' and Health_Problems = 'Accident';
926 • select * from patient_healthproblems_relation;
```
- Result Grid:** Shows the results of the last query, displaying three rows of data:

| Patient_ID | Health_Problems     |
|------------|---------------------|
| P23488     | Bone marrow failure |
| P23489     | Surgery             |
| P23490     | Aplastic anemia     |
| P23491     | Surgery             |
- Information Panel:** Shows the definition of the "patient" table:

```
Table: patient
Columns:
Patient_ID varchar(40) PK
Patient_Name varchar(40)
Gender varchar(10)
Nationality varchar(40)
Date_of_Birth date
Street varchar(40)
City varchar(40)
Province varchar(40)
```
- Status Bar:** Shows the weather as 31°C Mostly clear, system icons, and the date/time as 10:50 PM 3/31/2024.

### 3.5.29. Payment Table

MySQL Workbench Local instance MySQL80

Schemas: donor, patient, patient

```
944 • select * from payment;
945 • update payment set Amount = 50.0 where Payment_ID = 'PID8759301';
946 • update payment set Amount = 80.0 where Payment_ID = 'PID8759297';
947 • delete from payment where Payment_ID = 'PID8759284';
948 • select * from payment;
949 |
950
951
952
953
954
955
956
```

Table: patient

Columns:

| Patient_ID    | varchar(40) PK |
|---------------|----------------|
| Patient_Name  | varchar(40)    |
| Gender        | varchar(10)    |
| Nationality   | varchar(40)    |
| Date_of_Birth | date           |
| Street        | varchar(40)    |
| City          | varchar(40)    |
| Province      | varchar(40)    |

Result Grid | Filter Rows | Edit | Export/Import | Wrap Cell Content | Result Grid | Form Editor | Field Types | Query Stats | Apply | Revert

| Payment_ID | Amount | Paid_Date  | Patient_ID |
|------------|--------|------------|------------|
| PID8759297 | 80.0   | 2024-03-12 | P23489     |
| PID8759301 | 50.0   | 2024-03-15 | P23488     |
| PID8759342 | 20.0   | 2024-02-27 | P23490     |
| ...        | ...    | ...        | ...        |

payment 95 payment 96 x

Object Info Session Output

Query Completed

31°C Mostly clear

Search

10:50 PM 3/31/2024

# Chapter 4: Transactions

## 4.1. Simple Queries

### 4.1.1. Select Operation

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** blood\_bank\_management\_system
- Tables:** donor, patient, 01\_CreatingDataBase, donor, employee
- Query Editor:** Contains the SQL query: `-- Select Operation  
select * from DONOR where Gender = 'Male';`
- Result Grid:** Displays the results of the query, showing 4 rows of data for male donors.
- Action Output:** Shows the execution history of the query, including the time, action, message, and duration.
- System Bar:** Includes weather information (28°C, Partly sunny), system icons, and a taskbar with various application icons.

| Donor_ID | Donor_Name | Date_of_Birth | Nationality | Eligibility_Status | Blood_Quantity | Last_Donated_Date | Blood_Group | Street                  | City         | Province      | Blood_ID | Age  | Gender |
|----------|------------|---------------|-------------|--------------------|----------------|-------------------|-------------|-------------------------|--------------|---------------|----------|------|--------|
| D651354  | Dineth     | 2001-06-25    | Sinhalese   | 1                  | 390            | 2022-03-25        | O+          | Hilogama                | Kurunegala   | North Western | B33467   | 22   | Male   |
| D651362  | Yugan      | 2000-08-29    | Tamil       | 1                  | 385            | 2020-10-06        | O-          | Barathypuram            | Kilinochchi  | Northern      | B33492   | 23   | Male   |
| D651366  | Aravind    | 2001-12-01    | Tamil       | 1                  | 470            | 2022-05-27        | O+          | Parantan-Mullaitivu Hwy | Tannayuttu   | Northern      | B33493   | 22   | Male   |
| D651369  | Kavindu    | 1999-05-15    | Sinhalese   | 1                  | 430            | 2022-04-05        | O-          | Rupaha                  | Nuwara Eliya | Central       | B33500   | 24   | Male   |
| NULL     | NULL       | NULL          | NULL        | NULL               | NULL           | NULL              | NULL        | NULL                    | NULL         | NULL          | NULL     | NULL | NULL   |

### 4.1.2. Project Operation

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** blood\_bank\_management\_system
- Tables:** blood, blood\_bank, blood\_bloodtestsamples\_rel, blood\_branch\_relation, blood\_donation\_camp, blood\_employee\_relation, blood\_equipment\_relation, blood\_patient\_relation, bloodbank\_hospital\_relation, hranch\_availablebloodminn...
- Query Editor:** Contains the SQL query: `-- Project Operation  
select Employee_ID, Employee_Name, Nationality from employee;`
- Result Grid:** Displays the results of the query, showing 9 rows of data for employees.
- Action Output:** Shows the execution history of the query, including the time, action, message, and duration.
- System Bar:** Includes weather information (28°C, Mostly cloudy), system icons, and a taskbar with various application icons.

| Employee_ID | Employee_Name  | Nationality |
|-------------|----------------|-------------|
| E008        | Chamila        | Sinhalese   |
| E034        | Kamalan        | Sinhalese   |
| E045        | Keerthi Suresh | Sinhalese   |
| E053        | Kanthi         | Tamil       |
| E058        | Sunil          | Sinhalese   |
| E066        | Ranjith        | Sinhalese   |
| E072        | Pradeep        | Sinhalese   |
| E073        | Shyama         | Sinhalese   |
| NULL        | NULL           | NULL        |

### 4.1.3. Cartesian Product Operation

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Schemas list includes: blood\_patient\_relation, bloodbank\_hospital\_relation, branch\_availablebloodgroup, branch\_phonenumbers\_relation, donor, donor\_blooddonationcamp\_, donor\_donateddates\_relation, donor\_email\_relation, donor\_healthproblems\_relation, donor\_phonenumbers\_relation, employee, blood, hospital, blood\_bank.
- Query Editor:** Contains the following SQL code:

```
803
804
805
806
807
808
809 -- Cartesian Product Operation or cross product or cross join operation
810 • select * from patient cross join payment;
811
812
813
814
```
- Result Grid:** Displays the result of the query, showing 10 rows of data from the cross join between patient and payment tables. The columns are Patient\_ID, Patient\_Name, Gender, Nationality, Date\_of\_Birth, Street, City, Province, Age, Payment\_ID, Amount, Paid\_Date, and Patient\_ID.
- Table Information:** Shows the structure of the patient table with columns: Patient\_ID (PK), Patient\_Name, Gender, Nationality, Date\_of\_Birth, Street, City, Province, and Age.
- Object Info:** Shows the session information.
- System Bar:** Includes icons for file operations, search, and various tools.
- Taskbar:** Shows the date and time as 12:15 AM 4/4/2024.

### 4.1.4. Create a User View

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Schemas list includes: blood\_patient\_relation, bloodbank\_hospital\_relation, branch\_availablebloodgroup, branch\_phonenumbers\_relation, donor, donor\_blooddonationcamp\_, donor\_donateddates\_relation, donor\_email\_relation, donor\_healthproblems\_relation, donor\_phonenumbers\_relation, employee, blood, hospital, blood\_bank.
- Query Editor:** Contains the following SQL code:

```
809
810 -- Create a user view
811 • create view UV1 as select Hospital_ID, Hospital_Name, Province from Hospital;
812 • select * from UV1;
813
814
815
816
817
818
819
820
821
822
823
```
- Result Grid:** Displays the data from the UV1 user view, which is a selection from the Hospital table. The columns are Hospital\_ID, Hospital\_Name, and Province.
- Table Information:** Shows the structure of the patient table with columns: Patient\_ID (PK), Patient\_Name, Gender, Nationality, Date\_of\_Birth, Street, City, Province, and Age.
- Object Info:** Shows the session information.
- System Bar:** Includes icons for file operations, search, and various tools.
- Taskbar:** Shows the date and time as 12:34 AM 4/4/2024.

## 4.1.5. Renaming Operation

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** blood\_bank\_management\_system
- Tables:** blood, blood\_bank, blood\_blood\_test\_samples\_relation, blood\_branch\_relation, blood\_donation\_camp, blood\_employee\_relation, blood\_equipment\_relation, blood\_patient\_relation, bloodbank\_hospital\_relation, branch\_availablebloodgroups\_relation, branch\_phonenumbers\_relation.
- Code Editor:** Contains the following SQL code:

```
189 -- Renaming Operation
190 • show tables;
191 • rename table blood_bloodtestsamples_relation to blood_blood_test_samples_relation;
192 • show tables;
```
- Result Grid:** Shows the results of the SHOW TABLES command, listing the same 14 tables as the schema browser.
- Toolbar:** Includes standard MySQL Workbench icons for file operations, database management, and scripting.
- System Tray:** Shows the date and time as 12:36 AM 4/4/2024.

## 4.1.6. Use of an aggregation function

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** blood\_bank\_management\_system
- Tables:** employee\_email\_relation, employee\_phonenumbers\_relation, equipment, hospital, hospital\_phonenumbers\_relation, patient, patient\_email\_relation, patient\_healthproblems\_relation, patient\_phonenumbers\_relation, payment.
- Code Editor:** Contains the following SQL code:

```
189
190
191
192
193
194 -- Use of an aggregation function
195 • SELECT AVG(salary) AS AverageSalary, SUM(salary) AS TotalSalary, MIN(salary) AS LowestSalary,
196 MAX(salary) AS HighestSalary,COUNT(employee_id) AS NumberOfEmployees
197 FROM employee GROUP BY Branch_ID;
```
- Result Grid:** Shows the results of the aggregation query, displaying the following data:

| AverageSalary | TotalSalary | LowestSalary | HighestSalary | NumberOfEmployees |
|---------------|-------------|--------------|---------------|-------------------|
| 40000.0000    | 40000       | 40000        | 40000         | 1                 |
| 50000.0000    | 50000       | 50000        | 50000         | 1                 |
| 80000.0000    | 80000       | 80000        | 80000         | 1                 |
| 90000.0000    | 90000       | 90000        | 90000         | 1                 |
| 70000.0000    | 140000      | 60000        | 80000         | 2                 |
| 100000.0000   | 100000      | 100000       | 100000        | 1                 |
| 150000.0000   | 150000      | 150000       | 150000        | 1                 |
- Toolbar:** Includes standard MySQL Workbench icons for file operations, database management, and scripting.
- System Tray:** Shows the date and time as 12:57 AM 4/4/2024.

#### **4.1.7. Use of like keyword**

The screenshot shows the MySQL Workbench application window. The title bar reads "MySQL Workbench" and "Local instance MySQL80". The menu bar includes File, Edit, View, Query, Database, Server, Tools, Scripting, and Help. The toolbar contains various icons for database navigation and management. The left sidebar has sections for Navigator, Schemas, and Administration, with "Schemas" currently selected. The main workspace displays a query editor titled "Blood Bank Management System" with tabs for "donor", "01.CreatingDataBase", "payment", and "employee". The "payment" tab is active, showing a list of rows from the "payment" table. A SQL query is also present in the editor:

```
819  
820  
821  
822  
823 -- Use of LIKE keyword  
824 • select * from DONOR where Donor_Name like '%y%' or Donor_Name like '_yak%';  
825  
826  
827  
828  
829  
830  
831  
832  
833 ----- Complex queries -----
```

The results grid for the "payment" table shows two rows:

| Payment_ID | Amount | Paid_Date  | Patient_ID | Donor_ID | Donor_Name | Date_of_Birth | Nationality | Eligibility_Status | Blood_Quantity | Last_Donated_Date | Blood_Group | Street           | City        | Province | Blood_ID | Age | Gender |
|------------|--------|------------|------------|----------|------------|---------------|-------------|--------------------|----------------|-------------------|-------------|------------------|-------------|----------|----------|-----|--------|
| D651357    | 500.00 | 2023-12-04 | Kavya      | D651357  | Kavya      | 2002-11-11    | Tamil       | 1                  | 450            | 2023-12-04        | B+          | Thirunager North | Kilinochchi | Northern | B33470   | 21  | Female |
| D651362    | 500.00 | 2020-10-06 | Yugan      | D651362  | Yugan      | 2000-08-29    | Tamil       | 1                  | 385            | 2020-10-06        | O-          | Barathypuram     | Kilinochchi | Northern | B33492   | 23  | Male   |

Below the results grid, there are tabs for "DONOR 28", "Output", and "Action Output". The bottom status bar shows the time as 12:59 AM and the date as 4/4/2024.

## 4.2. Complex Queries

### 4.2.1. Basic set operations without user views

#### 1. Union Operation

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** The current schema is "donor". Other schemas listed include "branch\_phonenumbers\_relation", "donor", "donor\_blooddonationcamp\_relation", "donor\_donateddates\_relation", "donor\_email\_relation", "donor\_healthproblems\_relation", "donor\_phonenumbers\_relation", "employee", "employee\_donor\_relation", "employee\_email\_relation", "employee\_phonenumbers\_relation", "equipment", and "hospital".
- Code Editor:** The code editor displays a query for performing a Union operation:

```
833
834 -- ----- Complex queries -----
835
836 -- Union Operation
837 • (select Patient_ID, Patient_Name, Age from patient where City = 'Hatton')
838 union
839 (select Patient_ID, Patient_Name, Age from patient where City = 'Peliyagoda');
840
841
842
843
844
845
846
847
```

- Result Grid:** The result grid shows two rows of data from the union query:

| Patient_ID | Patient_Name | Age |
|------------|--------------|-----|
| P23489     | Ranima       | 48  |
| P23491     | Samtha       | -36 |

#### 2. Intersection Operation

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** The current schema is "donor". Other schemas listed include "patient", "donor\_phonenumbers\_relation", and "payment".
- Code Editor:** The code editor displays a query for performing an intersection operation using a natural join:

```
828
829 -- Intersection Operation
830 -- Since intersect keyword is not working in our version, we have used this query to perform intersection
831 • select * from Patient natural join payment
832 where Gender = 'Male' and Nationality = 'Tamil';
833
834
835
836
837
838
839
840
841
842
```

- Result Grid:** The result grid shows one row of data from the intersection query:

| Patient_ID | Patient_Name | Gender | Nationality | Date_of_Birth | Street   | City           | Province | Age | Payment_ID | Amount | Paid_Date  |
|------------|--------------|--------|-------------|---------------|----------|----------------|----------|-----|------------|--------|------------|
| P23490     | Pramitha     | Male   | Tamil       | 1970-12-24    | Street32 | Chavakachcheri | Northern | 53  | PID8759342 | 20.0   | 2024-02-27 |

### 3. Set Difference Operation

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `blood_bank_management_system` with various tables like `blood`, `blood_bank`, etc.
- SQL Editor:** Displays the following SQL code:

```
836 -- Set Difference Operation
837 • select * from DONOR where Province not in ('Northern', 'Central');
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
```
- Result Grid:** Shows the results of the query, listing donors from provinces other than Northern or Central. The columns include Donor\_ID, Donor\_Name, Date\_of\_Birth, Nationality, Eligibility\_Status, Blood\_Quantity, Last\_Donated\_Date, Blood\_Group, Street, City, Province, Blood\_ID, Age, and Gender. The data includes rows for Dinesh and Nimaya.
- Object Info:** Shows the table `bloodbank_hospital_relation` with columns `hospital_id` and `branch_id`.
- System Bar:** Shows the taskbar with various application icons and the system clock indicating 7:32 PM on 4/6/2024.

### 4. Division Operation

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `blood_bank_management_system` with various tables like `blood`, `blood_bank`, etc.
- SQL Editor:** Displays the following SQL code:

```
844
845 -- Division Operation
846 • SELECT p.Patient_ID, p.Patient_Name
847   FROM Patient p
848   WHERE NOT EXISTS (
849     SELECT m.Patient_ID
850     FROM payment as m
851   WHERE NOT EXISTS (
852     SELECT 1
853       FROM Patient as innerP
854      WHERE innerP.Patient_ID = m.Patient_ID
855   )
856 )
857
858
859
860
861
```
- Result Grid:** Shows the results of the query, listing patients who have not made any payments. The columns are Patient\_ID and Patient\_Name. The data includes rows for Kavindu Manuj, Pramitha, Ranima, and Samitha.
- Object Info:** Shows the table `bloodbank_hospital_relation` with columns `hospital_id` and `branch_id`.
- System Bar:** Shows the taskbar with various application icons and the system clock indicating 7:35 PM on 4/6/2024.

## 4.2.2. Relational Algebraic Operations by using user views

### 1. Inner Join

The screenshot shows the MySQL Workbench interface with the following details:

- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (blood\_bank\_management\_system), Tables (blood, blood\_bank, blood\_blood\_test\_samples\_relation, blood\_branch\_relation, blood\_donation\_center, blood\_employee\_relation, blood\_equipment\_relation, blood\_patient\_relation, bloodbank\_hospital\_relation, branch\_phonenumbers\_relation).
- Query Editor:** Title: Blood Bank Management System, Query Number: 01. Creating DataBase, Limit: 1000 rows. The code is:

```
-- -----Relational algebraic operations by using user view-----
-- Inner Join
create view UV3 as (
    select distinct Employee_ID, Employee_Name, Age from employee as e
    natural join blood_employee_relation as ber
    where ber.employee_id = e.Employee_ID
);
select * from UV3;
```

**Result Grid:** Shows the results of the query:

| Employee_ID | Employee_Name  | Age |
|-------------|----------------|-----|
| E045        | Keerthi Suresh | 40  |
| E053        | Kanthi         | 32  |
| E058        | Sunil          | 38  |
| E066        | Ranjith        | 55  |
| E072        | Pradeep        | 39  |

**Output:** Shows the execution log:

| Action      | Time   | Message                                      | Duration / Fetch      |
|-------------|--|--|-----------------------|
| 19 19:31:47 | SELECT * FROM blood_bank_management_system.blood LIMIT 0, 1000   | 7 row(s) returned                            | 0.000 sec / 0.000 sec |
| 20 19:32:03 | select * from DONOR where Province not in ('Northern', 'Central') LIMIT 0, 1000  | 2 row(s) returned                            | 0.000 sec / 0.000 sec |
| 21 19:35:03 | SELECT p.Patient_ID, p.Patient_Name FROM Patient p WHERE NOT EXISTS ( SELECT m.Patient_ID, m.Patient_Name FROM Patient m WHERE m.Patient_ID = p.Patient_ID ) | 4 row(s) returned                            | 0.047 sec / 0.000 sec |
| 22 19:35:19 | create view UV2 as ( select distinct Employee_ID, Employee_Name, Age from employee as e natural join blood_employee_relation as ber )                        | 0 row(s) affected                            | 0.062 sec             |
| 23 19:35:21 | create view UV2 as ( select distinct Employee_ID, Employee_Name, Age from employee as e natural join blood_employee_relation as ber )                        | Error Code: 1050. Table 'UV2' already exists | 0.000 sec             |
| 24 19:35:39 | create view UV2 as ( select distinct Employee_ID, Employee_Name, Age from employee as e natural join blood_employee_relation as ber )                        | Error Code: 1050. Table 'UV2' already exists | 0.000 sec             |

### 2. Natural Join

The screenshot shows the MySQL Workbench interface with the following details:

- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Schemas (blood\_bank\_management\_system), Tables (blood, blood\_bank, blood\_blood\_test\_samples\_relation, blood\_branch\_relation, blood\_donation\_center, blood\_employee\_relation, blood\_equipment\_relation, blood\_patient\_relation, bloodbank\_hospital\_relation, branch\_phonenumbers\_relation).
- Query Editor:** Title: Blood Bank Management System, Query Number: 01. Creating DataBase, Limit: 1000 rows. The code is:

```
-- Natural Join
create view UV4 as (
    select * from Hospital natural join bloodbank_hospital_relation
    where Province in ('North Western')
);
select * from UV4;
```

**Result Grid:** Shows the results of the query:

| Hospital_ID | Hospital_Name           | Street    | City       | Province      | branch_id |
|-------------|-------------------------|-----------|------------|---------------|-----------|
| H015        | Kurunegala Hospital     | Street#44 | Kurunegala | North Western | B021      |
| H016        | Chilaw General Hospital | Street#15 | Puttalam   | North Western | B037      |

### 3. Left Outer Join

The screenshot shows the MySQL Workbench interface with a query editor window titled "new---Blood\_Bank\_Management.donor". The code in the editor is:

```
848  
849  
850 -- Left Outer Join  
851 • create view UV4 as  
852 select * from DONOR as d left outer join BLOOD as b  
853 on b.Blood_ID = d.bloodId where Age = 23;  
854 ;  
855 • select * from UV4;  
856  
857  
858  
859  
860  
861  
862  
e2>
```

The result grid shows the following data:

| Donor_ID | Donor_Name | Date_of_Birth | Gender | Nationality | Eligibility_Status | Blood_Quantity | Last_Donated_Date | BloodGroup | Street       | City         | Province      | BloodId | Age | Blood_ID | Blood |
|----------|------------|---------------|--------|-------------|--------------------|----------------|-------------------|------------|--------------|--------------|---------------|---------|-----|----------|-------|
| D651355  | Tharushi   | 2000-09-05    | Female | Sinhalese   | 1                  | 500            | 2023-10-20        | A+         | Udasgiriya   | Matale       | Central       | B33469  | 23  | B33469   | A+    |
| D651360  | Nimaya     | 2001-03-18    | Female | Sinhalese   | 1                  | 400            | 2022-01-30        | A-         | Hurigasewwa  | Anuradhapura | North Central | B33486  | 23  | B33486   | A-    |
| D651362  | Yugan      | 2000-08-29    | Male   | Tamil       | 1                  | 385            | 2020-10-06        | O-         | Barathypuram | Kilinochchi  | Northern      | B33492  | 23  | B33492   | AB+   |

The output pane shows the following log entries:

| Action | Time     | Message  | Duration / Fetch      |
|--------|----------|--|-----------------------|
| 436    | 21:10:56 | create view UV4 as select * from DONOR as d left outer join BLOOD as b on b.Blood_ID = d.bloodId ... 0 row(s) affected | 0.016 sec             |
| 437    | 21:10:56 | select * from UV4 LIMIT 0, 1000 3 row(s) returned  | 0.000 sec / 0.000 sec |

### 4. Right Outer Join

The screenshot shows the MySQL Workbench interface with a query editor window titled "new---Blood\_Bank\_Management.donor". The code in the editor is:

```
864  
865  
866  
867 -- Right Outer Join  
868 • create view UV5 as  
869 select * from DONOR as d right outer join BLOOD as b  
870 on b.Blood_ID = d.bloodId where BloodGroup in ('A+', 'A-');  
871 • select * from UV5;  
872  
873  
874  
875  
876  
877  
878 e2>
```

The result grid shows the following data:

| Donor_ID | Donor_Name | Date_of_Birth | Gender | Nationality | Eligibility_Status | Blood_Quantity | Last_Donated_Date | BloodGroup | Street      | City         | Province      | BloodId | Age | Blood_ID | Blood |
|----------|------------|---------------|--------|-------------|--------------------|----------------|-------------------|------------|-------------|--------------|---------------|---------|-----|----------|-------|
| D651355  | Tharushi   | 2000-09-05    | Female | Sinhalese   | 1                  | 500            | 2023-10-20        | A+         | Udasgiriya  | Matale       | Central       | B33469  | 23  | B33469   | A+    |
| D651360  | Nimaya     | 2001-03-18    | Female | Sinhalese   | 1                  | 400            | 2022-01-30        | A-         | Hurigasewwa | Anuradhapura | North Central | B33486  | 23  | B33486   | A-    |

The output pane shows the following log entries:

| Action | Time     | Message   | Duration / Fetch      |
|--------|----------|---|-----------------------|
| 438    | 21:11:20 | create view UV5 as select * from DONOR as d right outer join BLOOD as b on b.Blood_ID = d.bloodId ... 0 row(s) affected | 0.031 sec             |
| 439    | 21:11:20 | select * from UV5 LIMIT 0, 1000 2 row(s) returned   | 0.000 sec / 0.000 sec |

## 5. Full Outer Join

The screenshot shows the MySQL Workbench interface with a query editor window titled "new --Blood\_Bank\_Management-- donor". The code is as follows:

```
881
882  -- Full Outer Join
883 •  create view UV6 as
884   select * from UV4
885   union
886   select * from UV5;
887 •  select * from UV6;
```

The results grid shows data from three tables: donor, UV4, and UV5. The columns are: Donor\_ID, Donor\_Name, Date\_of\_Birth, Gender, Nationality, Eligibility\_Status, Blood\_Quantity, Last\_Donated\_Date, BloodGroup, Street, City, Province, bloodId, Age, Blood\_ID, and Blood\_Group.

| Donor_ID | Donor_Name | Date_of_Birth | Gender | Nationality | Eligibility_Status | Blood_Quantity | Last_Donated_Date | BloodGroup | Street       | City         | Province      | bloodId | Age | Blood_ID | Blood_Group |
|----------|------------|---------------|--------|-------------|--------------------|----------------|-------------------|------------|--------------|--------------|---------------|---------|-----|----------|-------------|
| D651355  | Tharushi   | 2000-09-05    | Female | Sinhalese   | 1                  | 500            | 2023-10-20        | A+         | Udasgiriya   | Matale       | Central       | B33469  | 23  | B33469   | A+          |
| D651360  | Nimaya     | 2001-03-18    | Female | Sinhalese   | 1                  | 400            | 2022-01-30        | A-         | Hurigasivewa | Anuradhapura | North Central | B33486  | 23  | B33486   | A-          |
| D651362  | Yugan      | 2000-08-29    | Male   | Tamil       | 1                  | 385            | 2020-10-06        | O-         | Barathypuram | Kilinochchi  | Northern      | B33492  | 23  | B33492   | AB+         |

The output pane shows the creation of view UV6 and its execution.

## 6. Outer Union

The screenshot shows the MySQL Workbench interface with a query editor window titled "new --Blood\_Bank\_Management-- donor". The code is as follows:

```
893
894
895
896  -- Outer Union
897 •  create view UV7 as
898   select * from patient as p inner join payment as pt on p.Patient_ID = pt.PatientID
899   union
900   select Patient_ID, Patient_Name, Gender, Nationality, Date_of_Birth, Street, City, Province, Age, null, null, null, null from patient;
901 •  select * from UV7;
```

The results grid shows data from two tables: patient and payment. The columns are: Patient\_ID, Patient\_Name, Gender, Nationality, Date\_of\_Birth, Street, City, Province, Age, Payment\_ID, Amount, Paid\_Date, and PatientID.

| Patient_ID | Patient_Name  | Gender | Nationality | Date_of_Birth | Street   | City           | Province | Age | Payment_ID | Amount | Paid_Date  | PatientID |
|------------|---------------|--------|-------------|---------------|----------|----------------|----------|-----|------------|--------|------------|-----------|
| P23488     | Kavindu Manuj | Male   | Sinhalese   | 1995-11-09    | Street81 | Bandarawela    | Uva      | 28  | PID8759301 | 50.0   | 2024-03-15 | P23488    |
| P23489     | Ranima        | Female | Sinhalese   | 1976-03-21    | Street57 | Hatton         | Central  | 48  | PID8759297 | 80.0   | 2024-03-12 | P23489    |
| P23490     | Pramitha      | Male   | Tamil       | 1970-12-24    | Street32 | Chavakachcheri | Northern | 53  | PID8759342 | 20.0   | 2024-02-27 | P23490    |
| P23488     | Kavindu Manuj | Male   | Sinhalese   | 1995-11-09    | Street81 | Bandarawela    | Uva      | 28  | NULL       | NULL   | NULL       | NULL      |
| P23489     | Ranima        | Female | Sinhalese   | 1976-03-21    | Street57 | Hatton         | Central  | 48  | NULL       | NULL   | NULL       | NULL      |
| P23490     | Pramitha      | Male   | Tamil       | 1970-12-24    | Street32 | Chavakachcheri | Northern | 53  | NULL       | NULL   | NULL       | NULL      |
| P23491     | Samitha       | Female | Sinhalese   | 2061-01-30    | Street84 | Peliyagoda     | Western  | -36 | NULL       | NULL   | NULL       | NULL      |

The output pane shows the creation of view UV7 and its execution.

## 4.2.3. Nested Queries

### 1. Nested Query 1

The screenshot shows the MySQL Workbench interface with a query editor window titled "Blood Bank Management System". The code being run is:

```
917 -- Nested Queries
918
919 -- Nested Query 1
920 • select Donor_ID from Donor where (Age > 22) and Donor_ID
921 in (select donor_id from donor_blooddonationcamp_relation where camp_name in ('Camp6', 'Camp3')) ;
922
923
924
925
926
927
928
929
930
931
932
933
```

The result grid shows one row of data:

| Donor_ID |
|----------|
| D651362  |
| NULL     |

The output pane shows two log entries:

- 42 19:54:09 (select Patient\_ID, Patient\_Name, Age from patient where City = 'Hatton') union (select Patient\_ID, Pat... 2 row(s) returned
- 43 19:54:30 select Donor\_ID from Donor where (Age > 22) and Donor\_ID in (select donor\_id from donor\_blooddon... 1 row(s) returned

The system status bar at the bottom indicates the time is 7:54 PM on 4/6/2024.

### 2. Nested Query 2

The screenshot shows the MySQL Workbench interface with a query editor window titled "Blood Bank Management System". The code being run is:

```
935
936 -- Nested Query 2
937 • SELECT Branch_ID, Branch_Name FROM BLOOD_BANK
938 WHERE EXISTS (
939     SELECT 1
940     FROM EQUIPMENT
941     WHERE BLOOD_BANK.Branch_ID = EQUIPMENT.Branch_ID
942     AND Cost > 30000
943 );
944
945
946
947
948
949
950
951
```

The result grid shows multiple rows of data:

| Branch_ID | Branch_Name   |
|-----------|---------------|
| Br009     | Diyathalawa   |
| Br011     | Anuradhapura  |
| Br016     | Army Hospital |
| Br024     | Kilinochchi   |
| Br037     | Dambulla      |
| NULL      | NULL          |

The output pane shows one log entry:

- 43 19:54:30 select Donor\_ID from Donor where (Age > 22) and Donor\_ID in (select donor\_id from donor\_blooddon... 1 row(s) returned

The system status bar at the bottom indicates the time is 7:55 PM on 4/6/2024.

### 3. Nested Query 3

The screenshot shows the MySQL Workbench interface with a query editor and results grid.

**Query Editor:**

```
926
927
928
929    -- Nested Query 3
930 • select Patient_ID from patient where Age
931   in
932     (select Age from patient where age<40);
933
934
935
936
937
```

**Result Grid:**

| Patient_ID |
|------------|
| P23488     |
| P23491     |
| NULL       |

**Output:**

| Action Output | Time     | Action  | Message           | Duration / Fetch      |
|---------------|----------|---|-------------------|-----------------------|
| 443           | 21:11:49 | select * from UV7 LIMIT 0, 1000   | 7 row(s) returned | 0.000 sec / 0.000 sec |
| 444           | 21:12:03 | select Blood_ID from Blood where Blood_Group in (select BloodGroup from donor where Gender =Fe... | 3 row(s) returned | 0.000 sec / 0.000 sec |
| 445           | 21:12:37 | select Patient_ID from patient where Age in (select Age from patient where age<40) LIMIT 0, 1000  | 2 row(s) returned | 0.000 sec / 0.000 sec |

## Chapter 5: Tuning

The indexing technique is used to selected complex queries written in Chapter 4. In order to provide a clear understanding, the following steps are taken in query tuning for each complex query.

- Drop existing indices that are externally created in used tables
- Show indexes of table before creating suitable index (using SHOW INDEX command) to know about the existing indices.
- Show number of accessed rows before creating suitable index (using EXPLAIN command) when we are retrieving some particular data.
- Create suitable index in order to tune the query.
- Show indexes of table after creating suitable index (using SHOW INDEX command) to ensure that the index was created.
- Show the number of accessed rows after creating suitable index (using EXPLAIN command) when we are retrieving some particular data.

Tunning of a query is demonstrated in this chapter by comparing the number of accessed rows in the explain table before and after creating a suitable index. If the number of accessed rows decreases after the index is created, the query has been properly tuned. All of the screenshots included below meet that requirement.

### 5.1. Tuning of Union Operation

### 5.1.1. Before Tuning

```
MySQL 8.0 Command Line Client

mysql> select * from patient;
+-----+-----+-----+-----+-----+-----+-----+-----+
| Patient_ID | Patient_Name | Gender | Nationality | Date_of_Birth | Street | city | Province | Age |
+-----+-----+-----+-----+-----+-----+-----+-----+
| P23488 | Kavindu Manuj | Male | Sinhalese | 1995-11-09 | Street81 | Bandarawela | Uva | 28 |
| P23489 | Ranima | Female | Sinhalese | 1976-03-21 | Street57 | Hatton | Central | 48 |
| P23490 | Pramitha | Male | Tamil | 1970-12-24 | Street32 | Chavakachcheri | Northern | 53 |
| P23491 | Samitha | Female | Sinhalese | 2061-01-30 | Street84 | Peliyagoda | Western | -36 |
+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> show index from patient;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| patient | 0 | PRIMARY | 1 | Patient_ID | A | 5 | NULL | NULL | BTREE | | | YES | NULL |
| patient | 0 | Patient_Name | 1 | Patient_Name | A | 5 | NULL | NULL | BTREE | | | YES | NULL |
| patient | 0 | Patient_Name | 2 | Date_of_Birth | A | 5 | NULL | NULL | BTREE | | | YES | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> explain((select Patient_ID, Patient_Name, Age from patient where City = 'Hatton')union(select Patient_ID, Patient_Name, Age from patient where City = 'Peliyagoda'));
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY | patient | NULL | ALL | NULL | NULL | NULL | NULL | 4 | 25.00 | Using where |
| 2 | UNION | patient | NULL | ALL | NULL | NULL | NULL | NULL | 4 | 25.00 | Using where |
| 3 | UNION RESULT | <union1,2> | NULL | ALL | NULL | NULL | NULL | NULL | NULL | NULL | Using temporary |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set, 1 warning (0.00 sec)

mysql>
```

### 5.1.2. After Tuning

```
MySQL 8.0 Command Line Client

mysql> explain((select Patient_ID, Patient_Name, Age from patient where City = 'Hatton')union(select Patient_ID, Patient_Name, Age from patient where City = 'Peliyagoda'));
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY | patient | NULL | ALL | NULL | NULL | NULL | NULL | 4 | 25.00 | Using where |
| 2 | UNION | patient | NULL | ALL | NULL | NULL | NULL | NULL | 4 | 25.00 | Using where |
| 3 | UNION RESULT | <union1,2> | NULL | ALL | NULL | NULL | NULL | NULL | NULL | NULL | Using temporary |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set, 1 warning (0.00 sec)

mysql> create index city_Ind on PATIENT(City);
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> explain((select Patient_ID, Patient_Name, Age from patient where City = 'Hatton')union(select Patient_ID, Patient_Name, Age from patient where City = 'Peliyagoda'));
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY | patient | NULL | ref | city_Ind | city_Ind | 162 | const | 1 | 100.00 | NULL |
| 2 | UNION | patient | NULL | ref | city_Ind | city_Ind | 162 | const | 1 | 100.00 | NULL |
| 3 | UNION RESULT | <union1,2> | NULL | ALL | NULL | NULL | NULL | NULL | NULL | NULL | Using temporary |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set, 1 warning (0.00 sec)

mysql>
```

## 5.2. Tuning of Intersection Operation

### 5.1.3. Before Tuning

```
MySQL 8.0 Command Line Client
3 rows in set (0.00 sec)

mysql> select * from patient;
+-----+-----+-----+-----+-----+-----+-----+-----+
| Patient_ID | Patient_Name | Gender | Nationality | Date_of_Birth | Street | city | Province | Age |
+-----+-----+-----+-----+-----+-----+-----+-----+
| P23488 | Kavindu Manuj | Male | Sinhalese | 1995-11-09 | Street81 | Bandarawela | Uva | 28 |
| P23489 | Ranima | Female | Sinhalese | 1976-03-21 | Street57 | Hatton | Central | 48 |
| P23490 | Pramitha | Male | Tamil | 1970-12-24 | Street32 | Chavakachcheri | Northern | 53 |
| P23491 | Samitha | Female | Sinhalese | 2001-01-30 | Street84 | Peliyagoda | Western | -36 |
+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> show index from patient;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| patient | 0 | PRIMARY | 1 | Patient_ID | A | 5 | NULL | NULL | BTREE | | YES | NULL |
| patient | 0 | Patient_Name | 1 | Patient_Name | A | 5 | NULL | NULL | BTREE | | YES | NULL |
| patient | 0 | Patient_Name | 2 | Date_of_Birth | A | 5 | NULL | NULL | BTREE | | YES | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> explain(select * from Patient natural join payment where Gender = 'Male' and Nationality = 'Tamil');
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | Patient | NULL | ALL | NULL | NULL | NULL | NULL | 4 | 25.00 | Using where |
| 1 | SIMPLE | payment | NULL | ALL | NULL | NULL | NULL | NULL | 5 | 100.00 | Using join buffer (hash join) |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql>
```

#### 5.1.4. After Tuning

```
mysql> create index Nationality_Ind on PATIENT(Nationality);
Query OK, 0 rows affected (0.06 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> show index from patient;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| patient | 0 | PRIMARY | 1 | Patient_ID | A | 5 | NULL | NULL | BTREE | | YES | NULL |
| patient | 0 | Patient_Name | 1 | Patient_Name | A | 5 | NULL | NULL | BTREE | | YES | NULL |
| patient | 0 | Patient_Name | 2 | Date_of_Birth | A | 5 | NULL | NULL | BTREE | | YES | NULL |
| patient | 1 | Nationality_Ind | 1 | Nationality | A | 2 | NULL | NULL | BTREE | | YES | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

mysql> explain(select * from Patient natural join payment where Gender = 'Male' and Nationality = 'Tamil');
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | Patient | NULL | ref | Nationality_Ind | Nationality_Ind | 162 | const | 1 | 25.00 | Using where |
| 1 | SIMPLE | payment | NULL | ALL | NULL | NULL | NULL | NULL | 5 | 100.00 | Using join buffer (hash join) |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql>
```

## 5.3. Tuning of Set Difference Operation

### 5.1.5. Before Tuning

```
MySQL 8.0 Command Line Client

mysql> select * from donor;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Donor_ID | Donor_Name | Date_of_Birth | Gender | Nationality | Eligibility_Status | Blood_Quantity | Last_Donated_Date | BloodGroup | Street | city | Province
| bloodId | Age |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| D651355 | Tharushi | 2000-09-05 | Female | Sinhalese | 1 | 500 | 2023-10-20 | A+ | Udasgiriya | Matale | Central
| B33469 | 23 | Kavya | 2002-11-11 | Female | Tamil | 1 | 450 | 2023-12-04 | B+ | Thirunagar North | Kilinochchi | Northern
| D651357 | 21 | B33470 | Nimaya | 2001-03-18 | Female | Sinhalese | 1 | 400 | 2022-01-30 | A- | Hurigaswewa | Anuradhapura | North Central
| B33486 | 23 | D651362 | Yugan | 2000-08-29 | Male | Tamil | 1 | 385 | 2020-10-06 | O- | Barathypuram | Kilinochchi | Northern
| B33492 | 23 | D651366 | Aravind | 2001-12-01 | Male | Tamil | 1 | 470 | 2022-05-27 | O+ | Parantan-Mullaitivu Hwy | Tanniyuttru | Northern
| B33493 | 22 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> show index from donor;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| donor | 0 | PRIMARY | 1 | Donor_ID | A | 6 | NULL | NULL | BTREE | | YES | NULL |
| donor | 1 | fk_blood1 | 1 | bloodId | A | 6 | NULL | NULL | YES | BTREE | | YES | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> explain(select * from DONOR where Province not in ('Northern', 'Central'));
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | DONOR | NULL | ALL | NULL | NULL | NULL | NULL | 5 | 60.00 | Using where |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.00 sec)

mysql>
```

### 5.1.6. After Tuning

```
MySQL 8.0 Command Line Client

mysql> explain(select * from DONOR where Province not in ('Northern', 'Central'));
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | DONOR | NULL | ALL | NULL | NULL | NULL | NULL | 5 | 60.00 | Using where |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.00 sec)

mysql> create index Province_Ind on DONOR(Province);
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> show index from donor;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| donor | 0 | PRIMARY | 1 | Donor_ID | A | 6 | NULL | NULL | BTREE | | YES | NULL |
| donor | 1 | fk_blood1 | 1 | bloodId | A | 6 | NULL | NULL | YES | BTREE | | YES | NULL |
| donor | 1 | Province_Ind | 1 | Province | A | 3 | NULL | NULL | BTREE | | YES | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

mysql> explain(select * from DONOR where Province not in ('Northern', 'Central'));
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | DONOR | NULL | range | Province_Ind | Province_Ind | 202 | NULL | 3 | 100.00 | Using index condition |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.00 sec)

mysql>
```

## 5.4. Tuning of Natural Join Operation

### 5.1.7. Before Tuning

```
MySQL 8.0 Command Line Client

mysql> select * from UV3;
+-----+-----+-----+-----+
| Hospital_ID | Hospital_Name | Street | City | Province | branch_id |
+-----+-----+-----+-----+
| H015 | Kurunegala Hospital | StreetH4 | Kurunegala | North Western | Br021 |
| H016 | Chilaw General Hospital | StreetH5 | Puttalam | North Western | Br037 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> show index from UV3;
Empty set (0.00 sec)

mysql> show index from Hospital;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| hospital | 0 | PRIMARY | 1 | Hospital_ID | A | 7 | NULL | NULL | BTREE | | YES | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> show index from bloodbank_hospital_relation;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| bloodbank_hospital_relation | 0 | PRIMARY | 1 | hospital_id | A | 1 | NULL | NULL | BTREE | | YES | NULL |
| bloodbank_hospital_relation | 0 | PRIMARY | 2 | branch_id | A | 1 | NULL | NULL | BTREE | | YES | NULL |
| bloodbank_hospital_relation | 1 | fk_branch7 | 1 | branch_id | A | 1 | NULL | NULL | BTREE | | YES | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> explain(select * from UV3);
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | hospital | NULL | ALL | PRIMARY | NULL | NULL | NULL | 6 | 16.67 | Using where |
| 1 | SIMPLE | bloodbank_hospital_relation | NULL | ref | PRIMARY | PRIMARY | 162 | blood_bank_management_system.hospital.Hospital_ID | 1 | 100.00 | Using where; Using index |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql>
```

### 5.1.8. After Tuning

```
MySQL 8.0 Command Line Client

mysql> create index Province_Ind on Hospital(Province);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> show index from Hospital;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| hospital | 0 | PRIMARY | 1 | Hospital_ID | A | 7 | NULL | NULL | BTREE | | YES | NULL |
| hospital | 1 | Province_Ind | 1 | Province | A | 5 | NULL | NULL | BTREE | | YES | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)

mysql> show index from bloodbank_hospital_relation;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| bloodbank_hospital_relation | 0 | PRIMARY | 1 | hospital_id | A | 1 | NULL | NULL | BTREE | | YES | NULL |
| bloodbank_hospital_relation | 0 | PRIMARY | 2 | branch_id | A | 1 | NULL | NULL | BTREE | | YES | NULL |
| bloodbank_hospital_relation | 1 | fk_branch7 | 1 | branch_id | A | 1 | NULL | NULL | BTREE | | YES | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> explain(select * from UV3);
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | hospital | NULL | ref | PRIMARY,Province_Ind | Province_Ind | 202 | const | 2 | 100.00 | NULL |
| 1 | SIMPLE | bloodbank_hospital_relation | NULL | ref | PRIMARY | PRIMARY | 162 | blood_bank_management_system.hospital.Hospital_ID | 1 | 100.00 | Using where; Using index |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql>
```

## 5.5. Tuning of Right Outer Join Operation

### 5.1.9. Before Tuning

```
MySQL 8.0 Command Line Client
+-----+
8 rows in set (0.00 sec)

mysql> use blood_bank_management_system;
Database changed
mysql> select * from UV4;
+-----+
| Donor_ID | Donor_Name | Date_of_Birth | Gender | Nationality | Eligibility_Status | Blood_Quantity | Last_Donated_Date | BloodGroup | Street | City | Province | bloodId | Age | Blood_ID | Blood_Group |
+-----+
| D651355 | Tharushi | 2000-09-05 | Female | Sinhalese | 1 | 500 | 2023-10-20 | A+ | Udasgiriya | Matale | Central | B33469 | 23 | B33469 | A |
| 2024-07-11 | | | | | | 500 | stored | | | | | | | | |
| D651360 | Niwala | 2001-03-18 | Female | Sinhalese | 1 | 400 | 2022-01-30 | A- | Hurigasewa | Anuradhapura | North Central | B33486 | 23 | B33486 | A |
| 2024-05-20 | | | | | | 400 | stored | | | | | | | | |
| D651362 | Yugan | 2000-08-29 | Male | Tamil | 1 | 385 | 2020-10-06 | O- | Barathypuram | Kilinochchi | Northern | B33492 | 23 | B33492 | A |
| B+ | 2024-06-02 | | | | | 385 | stored | | | | | | | | |
+-----+
3 rows in set (0.00 sec)

mysql> show index from DONOR;
+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+
| donor | 0 | PRIMARY | 1 | Donor_ID | A | 6 | NULL | NULL | BTREE | | YES | NULL |
| donor | 1 | fk_blood1 | 1 | bloodId | A | 6 | NULL | NULL | YES | BTREE | | YES | NULL |
+-----+
2 rows in set (0.00 sec)

mysql> show index from BLOOD;
+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+
| blood | 0 | PRIMARY | 1 | Blood_ID | A | 2 | NULL | NULL | BTREE | | YES | NULL |
+-----+
1 row in set (0.00 sec)

mysql> explain(select * from UV4);
+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+
| 1 | SIMPLE | d | NULL | ALL | NULL | NULL | NULL | 5 | 20.00 | Using where |
| 1 | SIMPLE | b | NULL | eq_ref | PRIMARY | PRIMARY | 162 | blood_bank_management_system.d.bloodId | 1 | 100.00 | NULL |
+-----+
2 rows in set, 1 warning (0.00 sec)

mysql>
```

### 5.1.10. After Tuning

```
MySQL 8.0 Command Line Client
Query OK, 0 rows affected (0.03 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> create index Age on DONOR(Age);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> show index from DONOR;
+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+
| donor | 0 | PRIMARY | 1 | Donor_ID | A | 6 | NULL | NULL | BTREE | | YES | NULL |
| donor | 1 | fk_blood1 | 1 | bloodId | A | 6 | NULL | NULL | YES | BTREE | | YES | NULL |
| donor | 1 | Age | 1 | Age | A | 3 | NULL | NULL | YES | BTREE | | YES | NULL |
+-----+
3 rows in set (0.01 sec)

mysql> show index from BLOOD;
+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+
| blood | 0 | PRIMARY | 1 | Blood_ID | A | 2 | NULL | NULL | BTREE | | YES | NULL |
+-----+
1 row in set (0.00 sec)

mysql> explain(select * from UV4);
+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+
| 1 | SIMPLE | d | NULL | ref | Age | Age | 5 | const | 3 | 100.00 | NULL |
| 1 | SIMPLE | b | NULL | eq_ref | PRIMARY | PRIMARY | 162 | blood_bank_management_system.d.bloodId | 1 | 100.00 | NULL |
+-----+
2 rows in set, 1 warning (0.00 sec)

mysql>
```

## 5.6. Tuning of Left Outer Join Operation

### 5.1.11. Before Tuning

```
MySQL 8.0 Command Line Client
mysql> select * from UV5;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Donor_ID | Donor_Name | Date_of_Birth | Gender | Nationality | Eligibility_Status | Blood_Quantity | Last_Donated_Date | BloodGroup | Street | City | Province | bloodId | Age | Blood_ID | Bl
ood_Group | Expiry_Date | Blood_Quantity_ml | Blood_Description |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| D651355 | Tharushi | 2000-09-05 | Female | Sinhalese | 1 | 500 | 2023-10-20 | A+ | Udasgiriya | Matale | Central | B33469 | 23 | B33469 | A+
| D651360 | Nimaya | 2001-03-18 | Female | Sinhalese | 1 | 400 | 2022-01-30 | A- | Hurigasewa | Anuradhapura | North Central | B33486 | 23 | B33486 | A-
| D651360 | Nimaya | 2024-05-20 | Female | Sinhalese | 1 | 400 | stored | A- | Hurigasewa | Anuradhapura | North Central | B33486 | 23 | B33486 | A-
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> show index from DONOR;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| donor | 0 | PRIMARY | 1 | Donor_ID | A | 6 | NULL | NULL | BTREE | | YES | NULL |
| donor | 1 | fk_blood1 | 1 | bloodId | A | 6 | NULL | YES | BTREE | | YES | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> show index from BLOOD;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| blood | 0 | PRIMARY | 1 | Blood_ID | A | 2 | NULL | NULL | BTREE | | YES | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> explain(select * from UV5);
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | d | NULL | ALL | fk_blood1 | NULL | NULL | NULL | 5 | 40.00 | Using where |
| 1 | SIMPLE | b | NULL | eq_ref | PRIMARY | PRIMARY | 162 | blood.bank_management_system.d.bloodId | 1 | 100.00 | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql>
```

### 5.1.12. After Tuning

```
MySQL 8.0 Command Line Client
mysql> create index BloodGroup_Ind on DONOR(BloodGroup);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> show index from DONOR;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| donor | 0 | PRIMARY | 1 | Donor_ID | A | 6 | NULL | NULL | BTREE | | YES | NULL |
| donor | 1 | fk_blood1 | 1 | bloodId | A | 6 | NULL | NULL | BTREE | YES | NULL | YES | NULL |
| donor | 1 | BloodGroup_Ind | 1 | BloodGroup | A | 5 | NULL | NULL | BTREE | | YES | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> show index from BLOOD;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| blood | 0 | PRIMARY | 1 | Blood_ID | A | 2 | NULL | NULL | BTREE | | YES | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> explain(select * from UV5);
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | d | NULL | range | fk_blood1,BloodGroup_Ind | BloodGroup_Ind | 14 | NULL | blood.bank.management_system.d.bloodId | 2 | 100.00 | Using index condition; Using where |
| 1 | SIMPLE | b | NULL | eq_ref | PRIMARY | PRIMARY | 162 | blood.bank.management_system.d.bloodId | 1 | 100.00 | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql>
```

## 5.7. Tuning of Full Outer Union Operation

### 5.1.13. Before Tuning

```
MySQL 8.0 Command Line Client
mysql> select * from UV4 union select * from UV5;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Donor_ID | Donor_Name | Date_of_Birth | Gender | Nationality | Eligibility_Status | Blood_Quantity | Last_Donated_Date | BloodGroup | Street | City | Province | bloodId | Age | Blood_ID | B |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| D651355 | Tharushi | 2000-09-05 | Female | Sinhalese | 1 | 500 | 2023-10-20 | A+ | Udasgiriya | Matale | Central | B33469 | 23 | B33469 | A |
| D651360 | Nimaya | 2001-03-18 | Female | Sinhalese | 1 | 400 | 2022-01-30 | A- | Hurigasewa | Anuradhapura | North Central | B33486 | 23 | B33486 | A |
| D651362 | Yugan | 2000-08-29 | Male | Tamil | 1 | 385 | 2020-10-06 | O- | Barathypuram | Kilinochchi | Northern | B33492 | 23 | B33492 | A |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> explain(select * from UV4 union select * from UV5);
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY | d | NULL | ALL | PRIMARY | NULL | NULL | NULL | 5 | 20.00 | Using where |
| 1 | PRIMARY | b | NULL | eq_ref | PRIMARY | PRIMARY | 162 | blood_bank_management_system.d.bloodId | 1 | 100.00 | NULL |
| 2 | UNION | d | NULL | ALL | fk_blood1 | NULL | NULL | NULL | 5 | 40.00 | Using where |
| 2 | UNION | b | NULL | eq_ref | PRIMARY | PRIMARY | 162 | blood_bank_management_system.d.bloodId | 1 | 100.00 | NULL |
| 3 | UNION RESULT | union1,2> | NULL | ALL | NULL | Using temporary |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set, 1 warning (0.00 sec)

mysql>
```

### 5.1.14. After Tuning

```
MySQL 8.0 Command Line Client
mysql> create index Age_Ind on DONOR(Age);
Query OK, 0 rows affected (0.08 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> create index BloodGroup_Ind on DONOR(BloodGroup);
Query OK, 0 rows affected (0.08 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> show index from DONOR;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| donor | 0 | PRIMARY | 1 | Donor_ID | A | 6 | NULL | NULL | BTREE | | YES | NULL |
| donor | 1 | fk_blood1 | 1 | bloodId | A | 6 | NULL | NULL | YES | BTREE | | YES | NULL |
| donor | 1 | Age_Ind | 1 | Age | A | 3 | NULL | NULL | YES | BTREE | | YES | NULL |
| donor | 1 | BloodGroup_Ind | 1 | BloodGroup | A | 5 | NULL | NULL | BTREE | | YES | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

mysql> explain(select * from UV4 union select * from UV5);
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY | d | NULL | ref | Age_Ind | Age_Ind | 5 | const | 3 | 100.00 | NULL |
| 1 | PRIMARY | b | NULL | eq_ref | PRIMARY | PRIMARY | 162 | blood_bank_management_system.d.bloodId | 1 | 100.00 | NULL |
| 2 | UNION | d | NULL | range | fk_blood1,BloodGroup_Ind | BloodGroup_Ind | 14 | NULL | 2 | 100.00 | Using index condition; Using where |
| 2 | UNION | b | NULL | eq_ref | PRIMARY | PRIMARY | 162 | blood_bank_management_system.d.bloodId | 1 | 100.00 | NULL |
| 3 | UNION RESULT | union1,2> | NULL | ALL | NULL | NULL | NULL | NULL | NULL | NULL | Using temporary |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set, 1 warning (0.00 sec)

mysql>
```

## 5.8. Tuning of Nested Query 1

### 5.1.15. Before Tuning

```
MySQL> select Blood_ID from Blood where Blood_Group in (select BloodGroup from donor where Gender ='Female');
+-----+
| Blood_ID |
+-----+
| B33469 |
| B33470 |
| B33486 |
+-----+
3 rows in set (0.00 sec)

mysql> explain(select Blood_ID from Blood where Blood_Group in (select BloodGroup from donor where Gender ='Female'));
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | <subquery> | NULL | ALL | NULL | NULL | NULL | NULL | 100.00 | NULL |
| 1 | SIMPLE | Blood | NULL | ALL | NULL | NULL | NULL | NULL | 6 | 16.67 | Using where; Using join buffer (hash join) |
| 2 | MATERIALIZED | donor | NULL | ALL | NULL | NULL | NULL | NULL | 5 | 20.00 | Using where |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set, 1 warning (0.00 sec)

mysql> show index from donor;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| donor | 0 | PRIMARY | 1 | Donor_ID | A | 6 | NULL | NULL | YES | BTREE | | | YES | NULL |
| donor | 1 | fk_blood1 | 1 | bloodId | A | 6 | NULL | NULL | YES | BTREE | | | YES | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

### 5.1.16. After Tuning

```
MySQL> create index BloodGroup_Ind on donor(BloodGroup);
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> create index Gender_Ind on Donor(Gender);
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> show index from donor;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| donor | 0 | PRIMARY | 1 | Donor_ID | A | 6 | NULL | NULL | YES | BTREE | | | YES | NULL |
| donor | 1 | fk_blood1 | 1 | bloodId | A | 6 | NULL | NULL | YES | BTREE | | | YES | NULL |
| donor | 1 | BloodGroup_Ind | 1 | BloodGroup | A | 5 | NULL | NULL | YES | BTREE | | | YES | NULL |
| donor | 1 | Gender_Ind | 1 | Gender | A | 2 | NULL | NULL | YES | BTREE | | | YES | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

mysql> select Blood_ID from Blood where Blood_Group in (select BloodGroup from donor where Gender ='Female');
+-----+
| Blood_ID |
+-----+
| B33469 |
| B33470 |
| B33486 |
+-----+
3 rows in set (0.00 sec)

mysql> explain(select Blood_ID from Blood where Blood_Group in (select BloodGroup from donor where Gender ='Female'));
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | Blood | NULL | ALL | NULL | NULL | NULL | NULL | 6 | 100.00 | NULL |
| 1 | SIMPLE | donor | NULL | ALL | NULL | BloodGroup_Ind,Gender_Ind | BloodGroup_Ind | 14 | 1 | blood_bank_management_system.Blood.Blood_Group |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql>
```

## 5.9. Tuning of Nested Query 2

### 5.1.17. Before Tuning

```
MySQL 8.0 Command Line Client
mysql> select Donor_ID, Donor_Name, Gender from donor where Age < All(select Age from donor where Gender = 'Male');
+-----+-----+-----+
| Donor_ID | Donor_Name | Gender |
+-----+-----+-----+
| D651357 | Kavya      | Female  |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> explain(select Donor_ID, Donor_Name, Gender from donor where Age < All(select Age from donor where Gender = 'Male'));
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY    | donor  |          | ALL  | NULL          |     |        |     | 5    | 66.67  | Using where |
| 2 | SUBQUERY   | donor  |          | ALL  | NULL          |     |        | ref | 5    | 28.00  | Using where |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql> show index from donor;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| donor | 0           | PRIMARY  | 1            | Donor_ID    | A          | 6           |          |       |       | BTREE    |          |          | YES     | NULL    |
| donor | 1           | fk_blood1| 1            | bloodId    | A          | 6           |          |       |       | BTREE    |          |          | YES     | NULL    |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

### 5.1.18. After Tuning

```
MySQL 8.0 Command Line Client
2 rows in set (0.00 sec)

mysql> create index Gender_Ind on Donor(Gender);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> show index from donor;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| donor | 0           | PRIMARY  | 1            | Donor_ID    | A          | 6           |          |       |       | BTREE    |          |          | YES     | NULL    |
| donor | 1           | fk_blood1| 1            | bloodId    | A          | 6           |          |       |       | BTREE    |          |          | YES     | NULL    |
| donor | 1           | Gender_Ind| 1            | Gender      | A          | 2           |          |       |       | BTREE    |          |          | YES     | NULL    |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

mysql> explain(select Donor_ID, Donor_Name, Gender from donor where Age < All(select Age from donor where Gender = 'Male'));
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | PRIMARY    | donor  |          | ALL  | NULL          |     |        |     | 5    | 66.67  | Using where |
| 2 | SUBQUERY   | donor  |          | ALL  | NULL          | ref | Gender_Ind | Gender_Ind | 42   | 100.00 | NULL    |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set, 1 warning (0.00 sec)

mysql>
```

## 5.10. Tuning of Nested Query 3

### 5.1.19. Before Tuning

```
MySQL 8.0 Command Line Client
mysql> select Patient_ID from patient where Age in(select Age from patient where age<40);
+-----+
| Patient_ID |
+-----+
| P23488    |
| P23491    |
+-----+
2 rows in set (0.00 sec)

mysql> explain (select Patient_ID from patient where Age in(select Age from patient where age<40));
+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra
+-----+
| 1 | SIMPLE     | patient | NULL      | ALL  | NULL          | NULL | NULL    | NULL | 4    | 100.00   | Using where
| 1 | SIMPLE     | patient | NULL      | ALL  | NULL          | NULL | NULL    | NULL | 4    | 25.00    | Using where; FirstMatch(patient); Using join buffer (hash join)
+-----+
2 rows in set, 1 warning (0.00 sec)

mysql> show index from patient;
+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression
+-----+
| patient | 0 | PRIMARY | 1 | Patient_ID | A           | 5           | NULL    | BTREE   | YES   | YES | NULL       |          | YES      | NULL
| patient | 0 | Patient_Name | 1 | Patient_Name | A           | 5           | NULL    | BTREE   | YES   | YES | NULL       |          | YES      | NULL
| patient | 0 | Patient_Name | 2 | Date_of_Birth | A           | 5           | NULL    | BTREE   | YES   | YES | NULL       |          | YES      | NULL
+-----+
3 rows in set (0.00 sec)

mysql>
```

### 5.1.20. After Tuning

```
MySQL 8.0 Command Line Client
mysql> create index Age_Ind on Patient(Age);
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> show index from patient;
+-----+
| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Expression
+-----+
| patient | 0 | PRIMARY | 1 | Patient_ID | A           | 5           | NULL    | BTREE   | YES   | YES | NULL       |          | YES      | NULL
| patient | 0 | Patient_Name | 1 | Patient_Name | A           | 5           | NULL    | BTREE   | YES   | YES | NULL       |          | YES      | NULL
| patient | 0 | Patient_Name | 2 | Date_of_Birth | A           | 5           | NULL    | BTREE   | YES   | YES | NULL       |          | YES      | NULL
| patient | 1 | Age_Ind    | 1 | Age        | A           | 4           | NULL    | YES    | BTREE   | YES   | NULL       |          | YES      | NULL
+-----+
4 rows in set (0.01 sec)

mysql> explain (select Patient_ID from patient where Age in(select Age from patient where age<40));
+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra
+-----+
| 1 | SIMPLE     | patient | NULL      | range | Age_Ind     | Age_Ind | 5      | NULL | 2    | 100.00   | Using where; Using index
| 1 | SIMPLE     | patient | NULL      | ref   | Age_Ind     | Age_Ind | 5      | blood_bank_management_system.patient.Age | 1    | 100.00   | Using index; FirstMatch(patient)
+-----+
2 rows in set, 1 warning (0.00 sec)

mysql>
```