CPCS241-Database I-Spring2022-Project

LIFE SOURCE

DB Design

Group No: 4

Student Name	Student Number
Ashwaq Khayat	
Jetana Abudawood	
Hadeel Althbiti	
Hasna Bukhari	
Rima Alghamdi	

Table of Contents

PART I: Analysis

1 Problem Definition and Data Requirements

1.1 Problem Description

Every two seconds, someone needs blood in surgery. Blood is essential to save lives. Therefore, blood donation centres (blood banks) were created to provide this urgent need. To facilitate the process of blood donation, we decided to create a database to help gather all needed information and arrange them in an efficient way that will ease access and store all required information.

The main purpose of establishing this blood bank database (DB) is to facilitate and manage the donations, reservations, and supplement of blood quantities required to patients, and this could be done by providing all the necessary data in an organized way that will ease the retrieval, modification, and addition operations. Hence, the blood bank workers can view precise information quickly.

The database is made to schedule donation appointments and patients' registrations, and collect all information about donors, donees, donations, the list of reservations, blood requests, and all related information to the donation process in general. Then organize and classify them to ease the management of blood transfusions between the blood bank and the needy. Also, the DBMS will speed the access and retrieval of required information. For example, displaying different records such as donation records, donor records, searching for a specific donor information, donation dates, information of the health practitioner who's responsible for the donation process and other services. In addition, it helps manage reservations accurately and professionally, and keeps track of blood bags requisitions and issuance.

Life Source DB will also be helpful for blood studies and would help the proper monitoring of blood available in the blood bank and easy processing of blood requests. With life source DB, the blood bank employees will not need to worry about the small analytic details that help them focus on their work.

1.2 Data Requirements

- **Blood Bank Entity:** each blood bank has unique name, address, phone number, number of employees, and email.

- **Hospital Entity:** each hospital has unique name and address, working hours, phone number, and number of employees.

- **Blood Stock Entity**: each stock has unique ID, stock type, stock date.

- **Donor Entity:** each donor has unique ID, BBCode, name(first name, last name), gender, birth date, age, weight, Hgb, blood pressure, temperature, phone number, blood type, Donor Status.

- **Donee Entity:** each donee has unique file number, name(first name, last name), address, gender, phone number, blood type, and birth date.

- **Donation weak Entity:** each donation has a type(Replacement or Volunteer), amount, method, name of Health Practitioner (Phlebotomist), Outcome, collection time, collection date, volume collection, weight, Hgb, blood pressure, temperature.

- **Requests:** each request has a unique ID, recipient ID (Donee's file number), date of request, blood type, quantity.

- **Reservation Entity:** each has a unique code, visit date, next visit date, time, and verification.

- **Employee Entity:** each employee has a unique ID, name(last name, first name), manager ID, address, birth date, email, phone, start date, qualification, and salary.
  - Health Practitioner_ Employee: each health practitioner must have a medical license.
  - Manager_ Employee: each manager must have work experience.
  - Receptionist _Employee Entity

- **Department Entity:** each department has a unique name, location(number of floor, building), number of patients, number of employees, and number of rooms.

- **Dependent Weak Entity:** each dependent has a relative relation, gender, name, phone number, age, birth date.

1.3 Business Rules

Employee:

- The employee must have a bachelor's degree.
- Each employee has a unique ID.

Health Practitioner Employee:

- Must have a verified medical certification.

Manager:

- A manager must have10 years of experience in the field.

Donee:

- Urgent need of blood.

Dependent:

- For males, the dependent age should be 18 or less.

Reservation:

- For a doner, the reservation date must be 8 weeks at least after the last donation.

Blood stock:

- The component may be stored for a maximum of 35 days.
- Each bag of blood must have Special ID.

Department:

- Each Department Should have a manager.

Donor:

- Age must be greater than 17 and less than 66 years.
- Each donor must have a unique Blood bank ID that starts with BB followed by 8 digits.
- Each donor has BBCode.

Donation:

- The donor donates to the blood bank.
- The donor can donate up to two bags in every donation.
- Weight must be at least 50 kg.
- Haemoglobin level must be between 13 and 17.
- The donors blood pressure should not be higher than 180/100 mmHg.
- The donors body temperature should not be higher than 37.5°C.

Blood bank:

- Each Blood bag is stored in blood stock.

Hospital:

- Each hospital has an address that consists of city and district.

1.4 Intended Output of the System

Output & Queries:

- Display max number of reservations by a donor.
- Display donation records with the ability to specify some properties.
- Display the unavailable blood types.
- Display Donor’s Information with visit dates.
- Display the verified donors before blood donation.
- Display donors' records.
- Display all booking records.
- Check for availability of quantities of blood types.
- Show the percentage of blood needed for each type.
- Show the amount of deficiency in stored blood.
- Display list of hospitals that submitted blood requests.

Transactions:

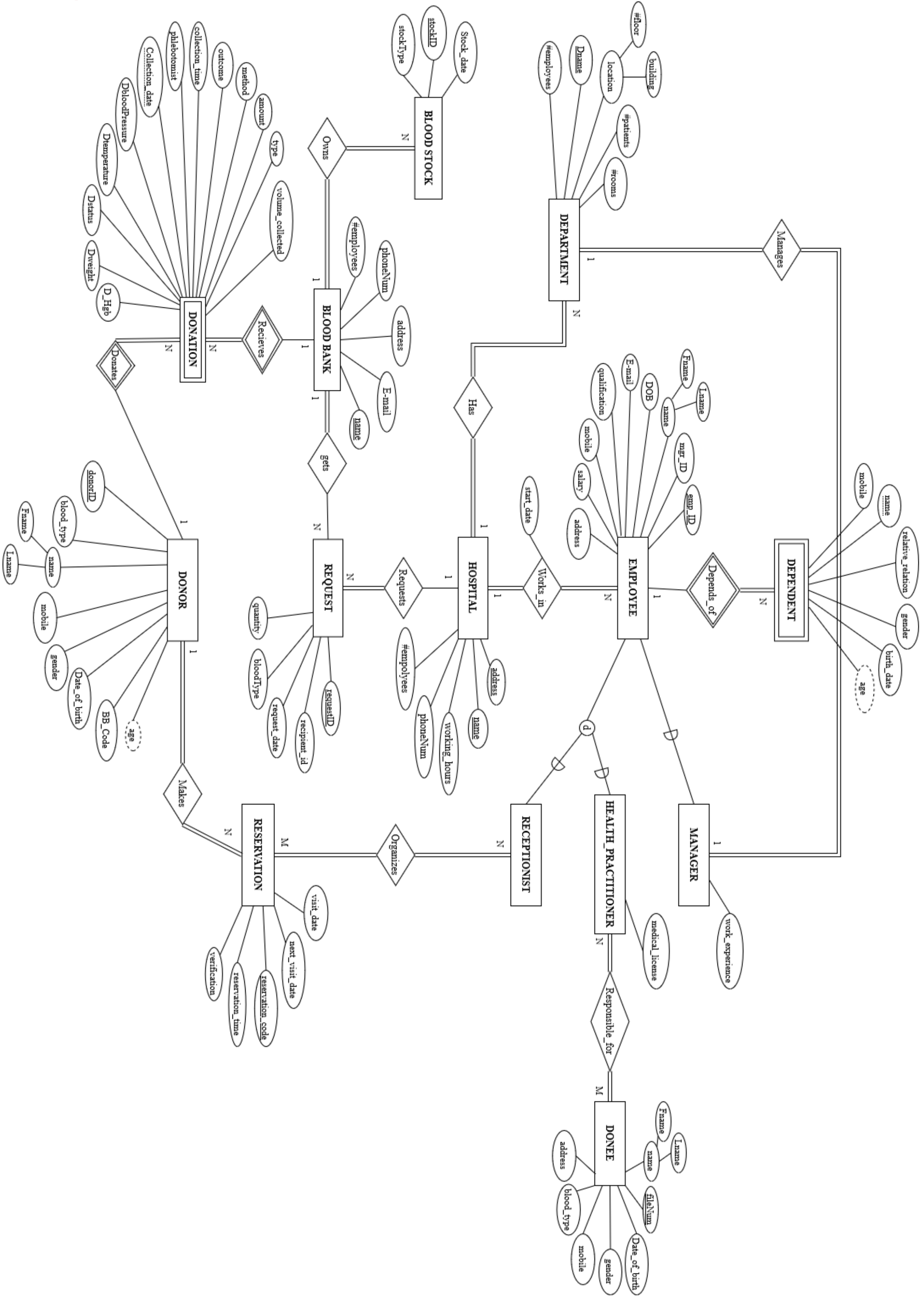
- Set a new reservation.
- Delete cancelled reservations.
- Search for donor information by his national ID.
- Search for quantities available in the stock of blood bags.
- Search for donation records.
- Search for order records.
- Calculate the percentage of monthly donations.
- Calculate stock shortage rate for all blood types & components.
- Calculate the percentage of requests from each hospital and their departments.

PART II: DB DEISGN

2 ER Diagram Design

2.1 ER Diagram

2.2 Design of Business Rules



Entity	Business Rule	Design Decisions	Justification (if any)
Employee	The employee must have a bachelor’s degree.	The Qualification attribute	
	Each employee has unique ID.	The emp_ID attribute.	A PK 🔑
Health Practitioner Employee	Must have a verified medical certification.	Medical_licence attribute	
Manager	A manager must have 10 years of experience in the field.	Work_expericence attribute	10 years have passed since the date of start work. Work_expericence >= 10 years
Donee	Urgent need of blood.	N:1 Weak binary relationship between the donee and health practitioner	Each donee nust has a health practitioner to report his need for blood to the hospital.
Reservation	For a doner, the reservation date must be 8 weeks at least after the last donation.	Next_Visit_Date attribute	Next_visit_date = reservation date + 8 weeks.
Blood Stock	The component may be stored for a maximum of 35 days.	Stock_date attribute	Stock_Date <= current date
	Each bag of blood must have Special ID.	Stock_ID attribute.	A PK 🔑

Department	Each Department Should have a manager.	1-1 Binary relationship, partial participation Between Employee entity and Department entity	Every department has a manager but not every employee is a manager.
Donor	Age must be greater than 17 and less than 66 years.	Age attribute	Domain of age (66>age>=17).
	Each donor must have a unique Blood bank ID that starts with BB followed by followed by 8 digits.	BB_code attribute	format "BB********" , a unique-candidate key. 🔑

Donation	Donor donates to the blood bank.	N:1 weak Binary relationship between Donation and Donor. N:1 weak Binary relationship between Donation and Blood Bank.	Each Donation must belong to one Donor. The blood bank collects the donations.
	Donor can donate up to two bags in every donation	Amount attribute	Check(2>=amount>=1)

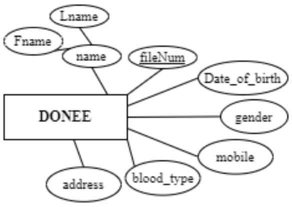
	Weight must be at least 50 kg.	Dweight attribute	Domain of Weight (weight >= 50)
	Haemoglobin level must be 13-17	D_Hgb attribute	Domain of HGB (17>=HGB>=13)
	Donors' blood pressure should not be higher than180/100 mmHg.	DbloodPressure attribute	Domain of Blood_pressure (180/100>=HGB)
	Donors' body temperature should not be higher than37.5°C.	Dtemperature attribute	Domain of temperature (37.5>= temperature)
	donor status must be valid to be able to donate	Dstatus attribute	Check (status="valid")
Blood bank	Each Blood bag is stored in bloodstock.	N-1 Binary relationship, partial participation, Between Blood Bank and Blood stock	Each Blood Bank Stored the blood in more than one Blood Stock.
Hospital	Each hospital has an address that consists of the province and the city	Address attribute	

3 ER-to-logical Schema Mapping

3.1 Mapping of Regular Entity Types

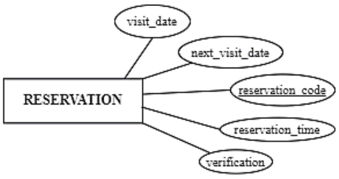
Donee

Fname	Lname	FileNum	Blood_Type	Address	Gender	Mobile	Date_of_birth



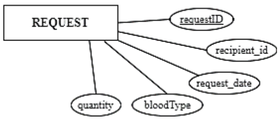
Resevation

Reservation_Code	Visit_Date	Next_Visit_Date	Reservation_Time	Verification	Donor_ID



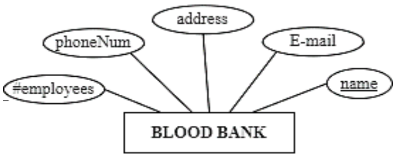
Request

RequestID	Recipient_ID	Request_Date	BloodType	BloodBank_Name	Hospital_Address	Hospital_Name	Quantity



Blood Bank

Name	Address	PhoneNum	#Employee	E-mail



Employee

Fname	Lname	emp_ID	Salary	E-mail	Mobile	Start_Date	DOB	Qualification	Mgr_ID	Address	Hospital_Name	Hospital_Address

Manager

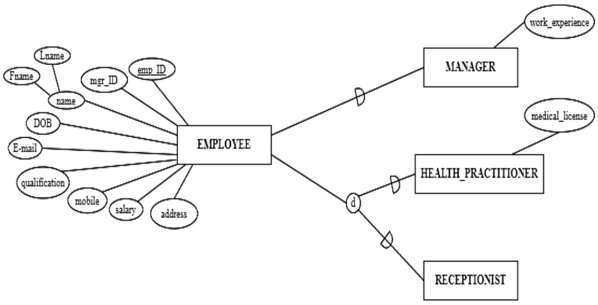
work_experience	emp_ID

Receptionist

emp_ID

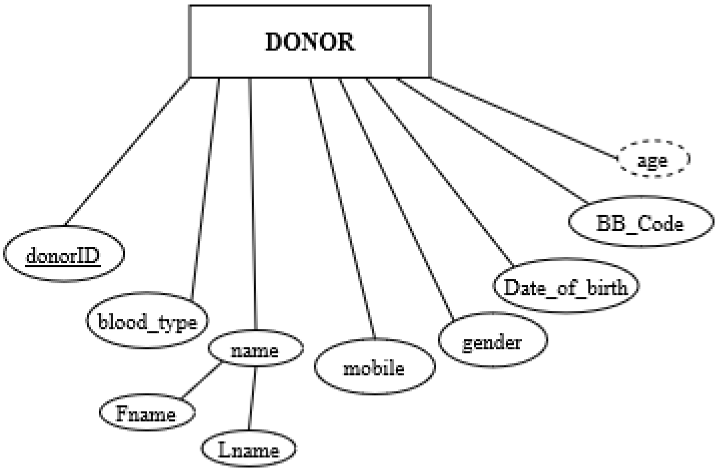
Health_Practitioner

emp_ID	Medical_License



Donor

DonorID	BB_code	Fname	Lname	Age	Gender	Blood_Type	mobile	DOB



3.2 Mapping of Weak Entity Types

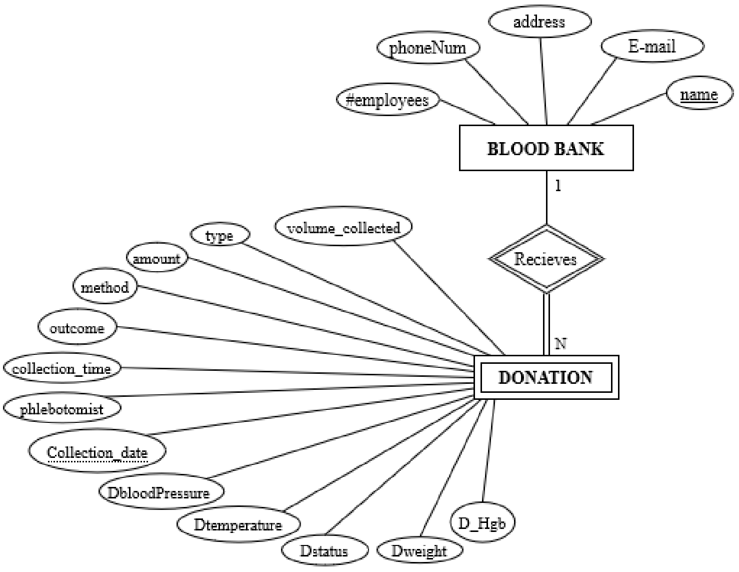
We chose the entities (Donation, Dependent) to be weak entities since they won't exist if their owner entities were not existed. Also, they don't have any special attributes that could be considered as primary keys, so we use partial keys instead.

Donation

Collection_Date	DonorID	BloodBank_name	Method	Amount	Type	Outcome	Phlebotomist	Collection_Time	Volume_collected	Dweight	Dstatus	Dtemperature	D_Hgb	DbloodPressure

Blood Bank

Name	Address	PhoneNum	#Employee	E-mail

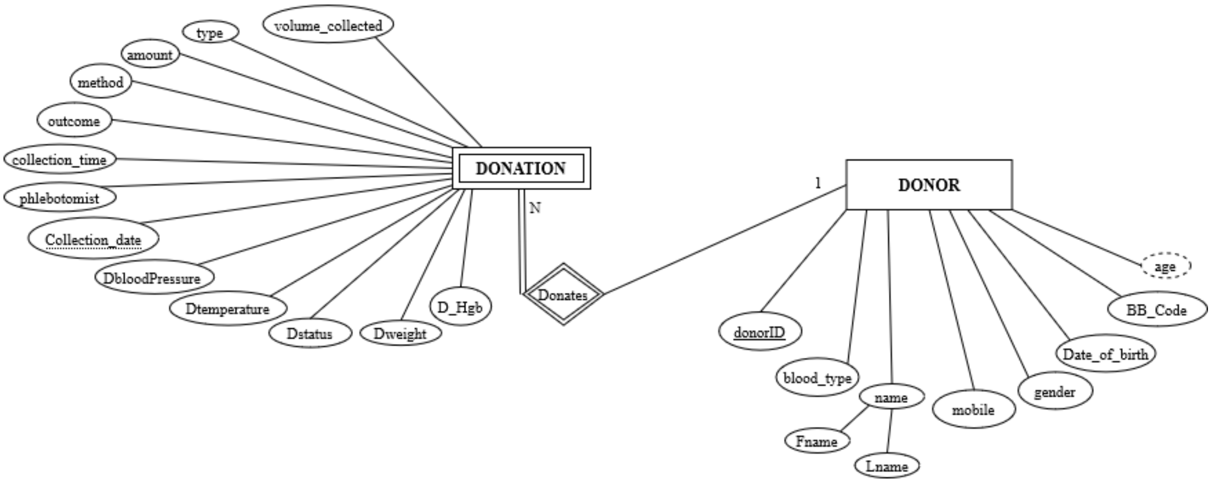


Donation

Collection_Date	DonorID	BloodBank_name	Method	Amount	Type	Outcome	Phlebotomist	Collection_Time	Volume_collected	Dweight	Dstatus	Dtemperature	D_Hgb	DbloodPressure

Donor

DonorID	BB_code	Fname	Lname	Age	Gender	Blood_Type	mobile	DOB

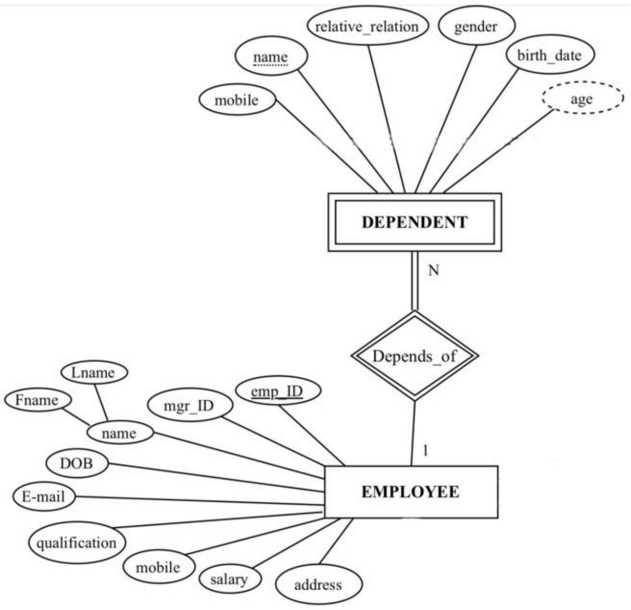


Employee

Fname	Lname	emp_ID	Salary	E-mail	Mobile	Start_Date	DOB	Qualification	Mgr_ID	Address	Hospital_Name	Hospital_Address

Dependent

emp_id	Name	Relative_Relation	Gender	Mobile	Birth_date	Age



3.3 Mapping of Binary 1-1 Relationship Types

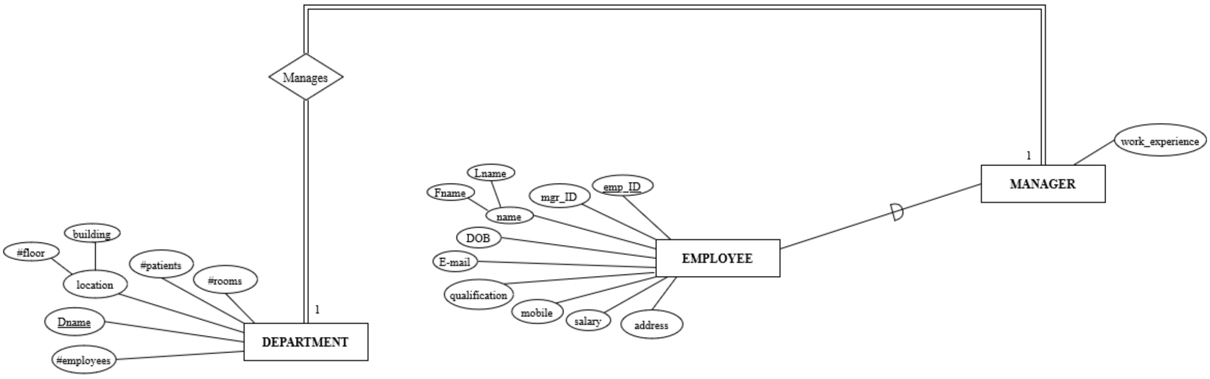
When the relationship is limited to only one element of each party of the relationship, where each element of the first party is related to only one element of the other, then the most appropriate cardinality ratio is one-to-one 1:1.

Department

Dname	#Rooms	#Patients	#Employees	Floor_num	Building	Hospital_Name	Hospital_Address	Mgr_ID

Manager

work_experience	emp_ID



3.4 Mapping of Binary 1-N Relationship Types

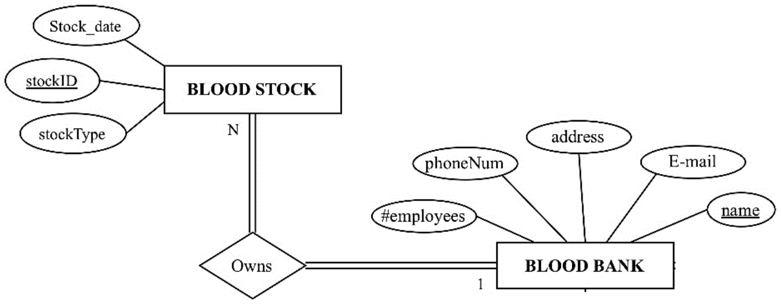
When the relationship makes connection between one element of the first party with several elements of the second one but not vice versa, the most appropriate cardinality ratio is one-to-many 1:N.

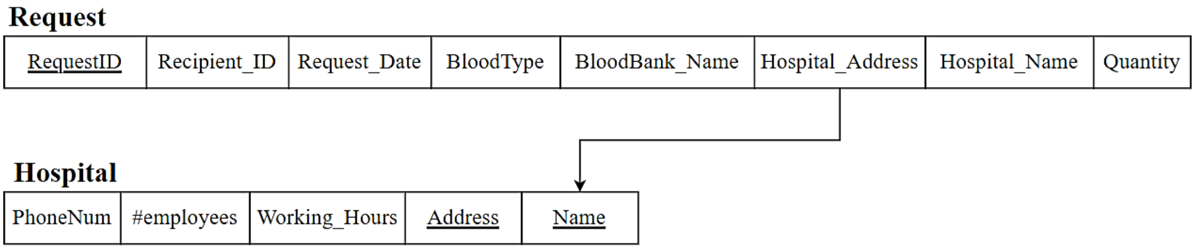
Blood Bank

Name	Address	PhoneNum	#Employee	E-mail

Blood Stock

StockID	Stock_Type	BloodBank_Name	Stock_Date



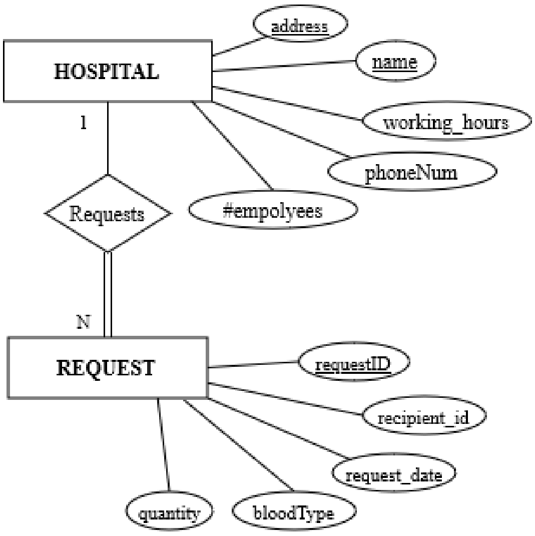


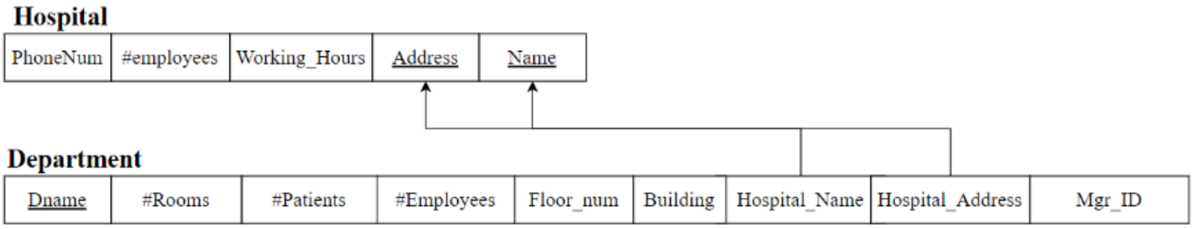
Request

RequestID	Recipient_ID	Request_Date	BloodType	BloodBank_Name	Hospital_Address	Hospital_Name	Quantity

Hospital

PhoneNum	#employees	Working_Hours	Address	Name



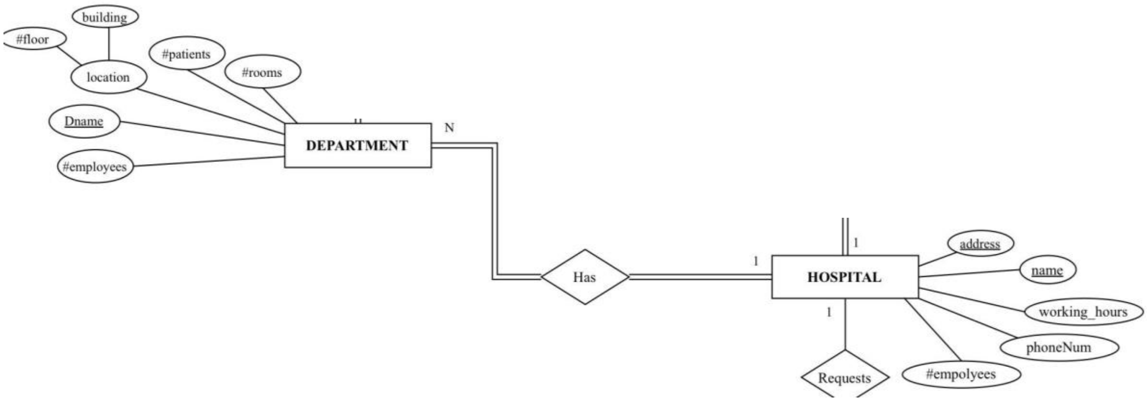


Hospital

PhoneNum	#employees	Working_Hours	Address	Name

Department

Dname	#Rooms	#Patients	#Employees	Floor_num	Building	Hospital_Name	Hospital_Address	Mgr_ID

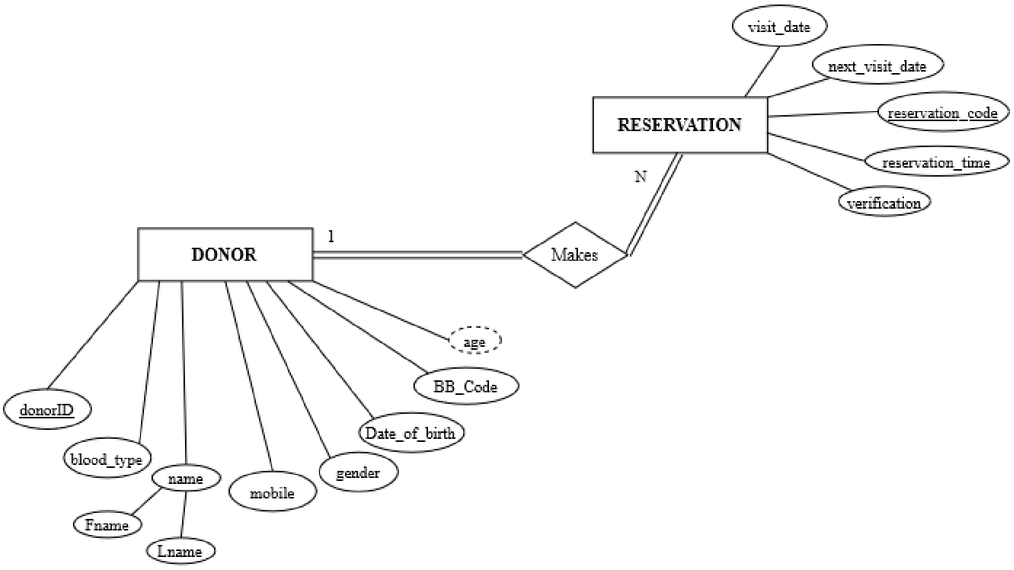


Donor

DonorID	BB_code	Fname	Lname	Age	Gender	Blood_Type	mobile	DOB

Resevation

Reservation_Code	Visit_Date	Next_Visit_Date	Reservation_Time	Verification	Donor_ID

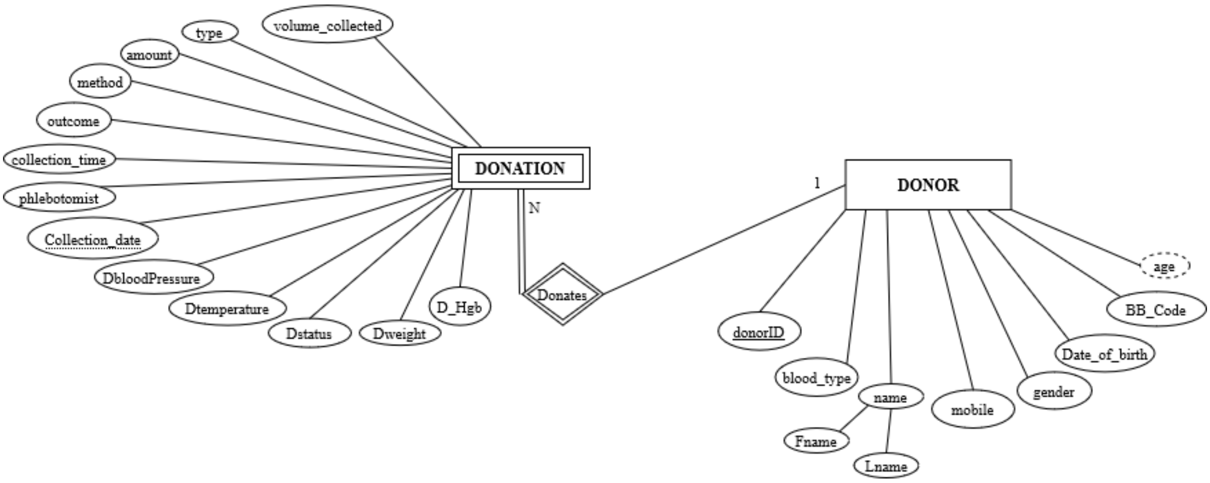


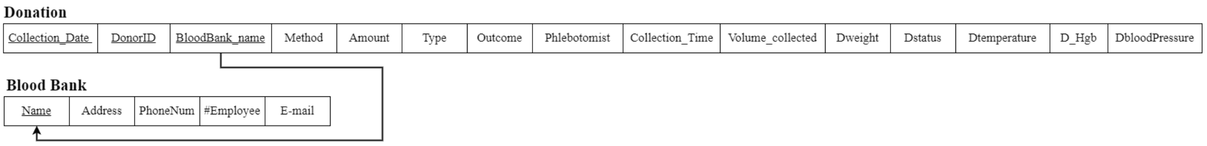
Donation

Collection_Date	DonorID	BloodBank_name	Method	Amount	Type	Outcome	Phlebotomist	Collection_Time	Volume_collected	Dweight	Dstatus	Dtemperature	D_Hgb	DbloodPressure

Donor

DonorID	BB_code	Fname	Lname	Age	Gender	Blood_Type	mobile	DOB



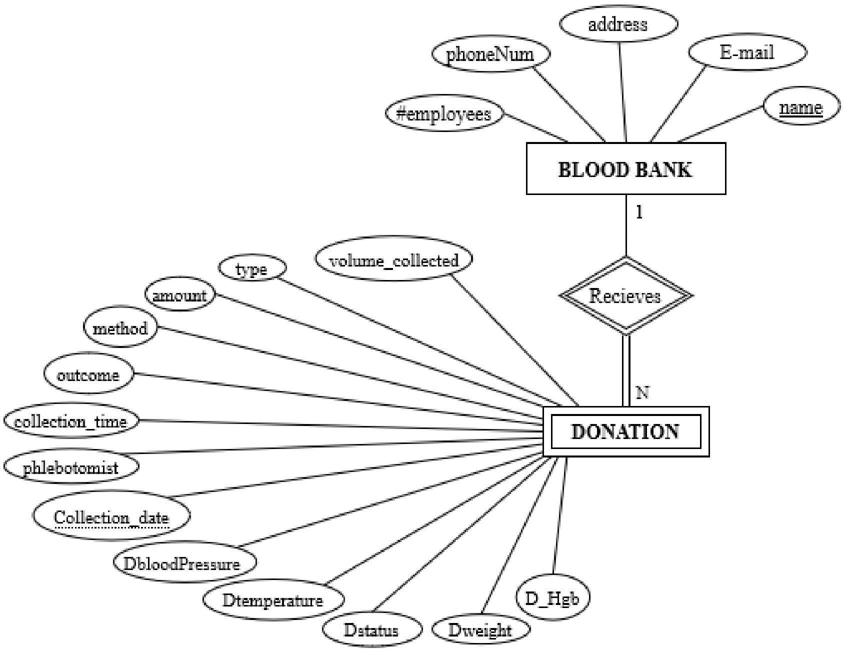


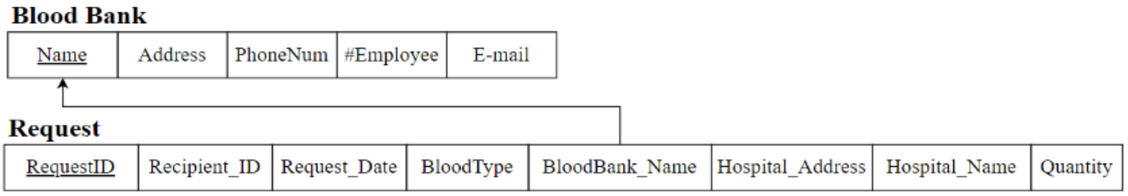
Donation

Collection_Date	DonorID	BloodBank_name	Method	Amount	Type	Outcome	Phlebotomist	Collection_Time	Volume_collected	Dweight	Dstatus	Dtemperature	D_Hgb	DbloodPressure

Blood Bank

Name	Address	PhoneNum	#Employee	E-mail



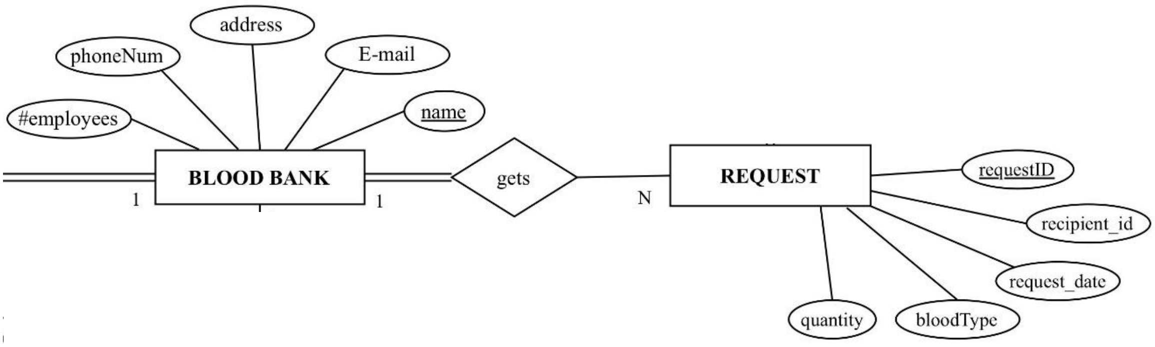


Blood Bank

Name	Address	PhoneNum	#Employee	E-mail

Request

RequestID	Recipient_ID	Request_Date	BloodType	BloodBank_Name	Hospital_Address	Hospital_Name	Quantity

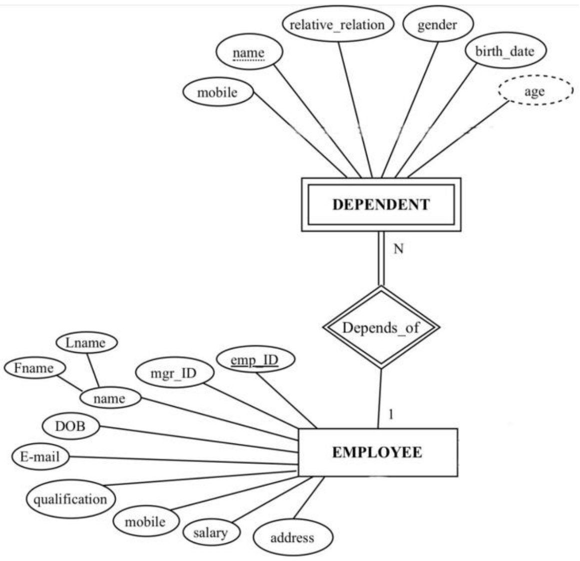


Employee

Fname	Lname	emp_ID	Salary	E-mail	Mobile	Start_Date	DOB	Qualification	Mgr_ID	Address	Hospital_Name	Hospital_Address

Dependent

emp_id	Name	Relative_Relation	Gender	Mobile	Birth_date	Age

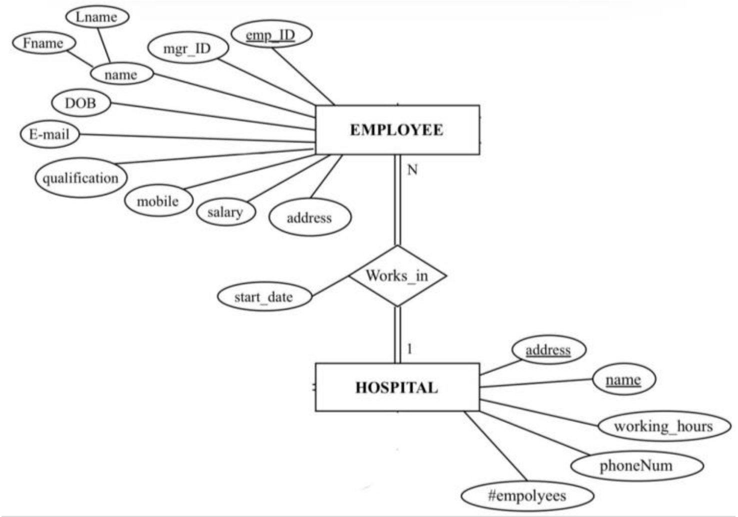


Hospital

PhoneNum	#employees	Working_Hours	Address	Name

Employee

Fname	Lname	emp_ID	Salary	E-mail	Mobile	Start_Date	DOB	Qualification	Mgr_ID	Address	Hospital_Name	Hospital_Address



3.5 Mapping of Binary M-N Relationship Types

The relationship that links several elements of the first party with many elements of the second, since an element of the first party may be related to several elements of the second party and vice versa, this relationship is considered as many-to-many N:M.

Responsible For

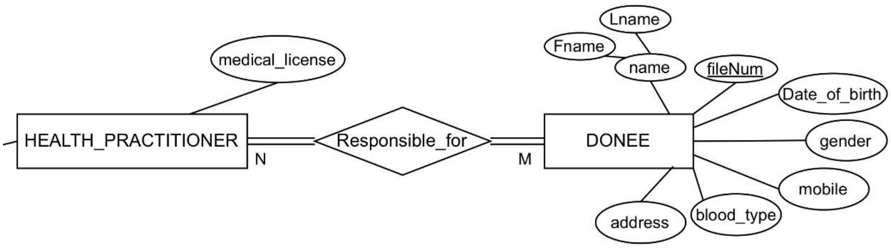
emp_ID	FileNum

Health_Practitioner

emp_ID	Medical_License

Donee

Fname	Lname	FileNum	Blood_Type	Address	Gender	Mobile	Date_of_birth



Resevation

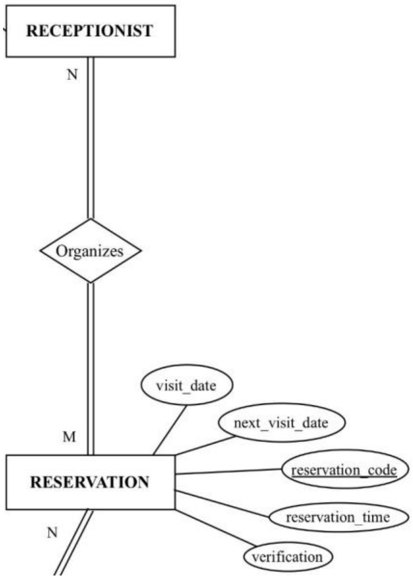
Reservation_Code	Visit_Date	Next_Visit_Date	Reservation_Time	Verification	Donor_ID

Organizes

emp_ID	Reservation_Code

Receptionist

emp_ID



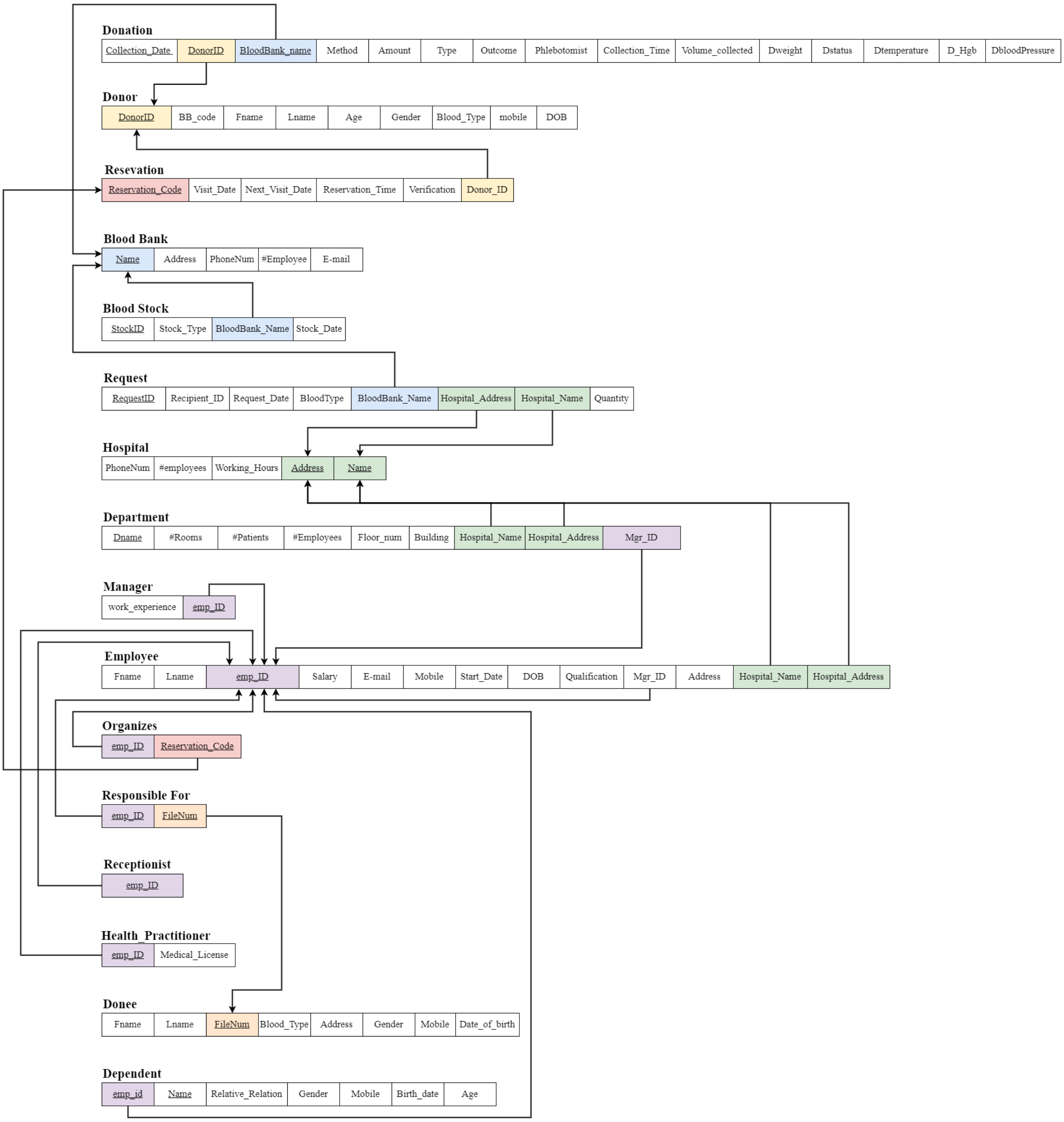
3.6 Mapping of Multivalued Attributes

None

3.7 Mapping of N-ary Relationship Types

None

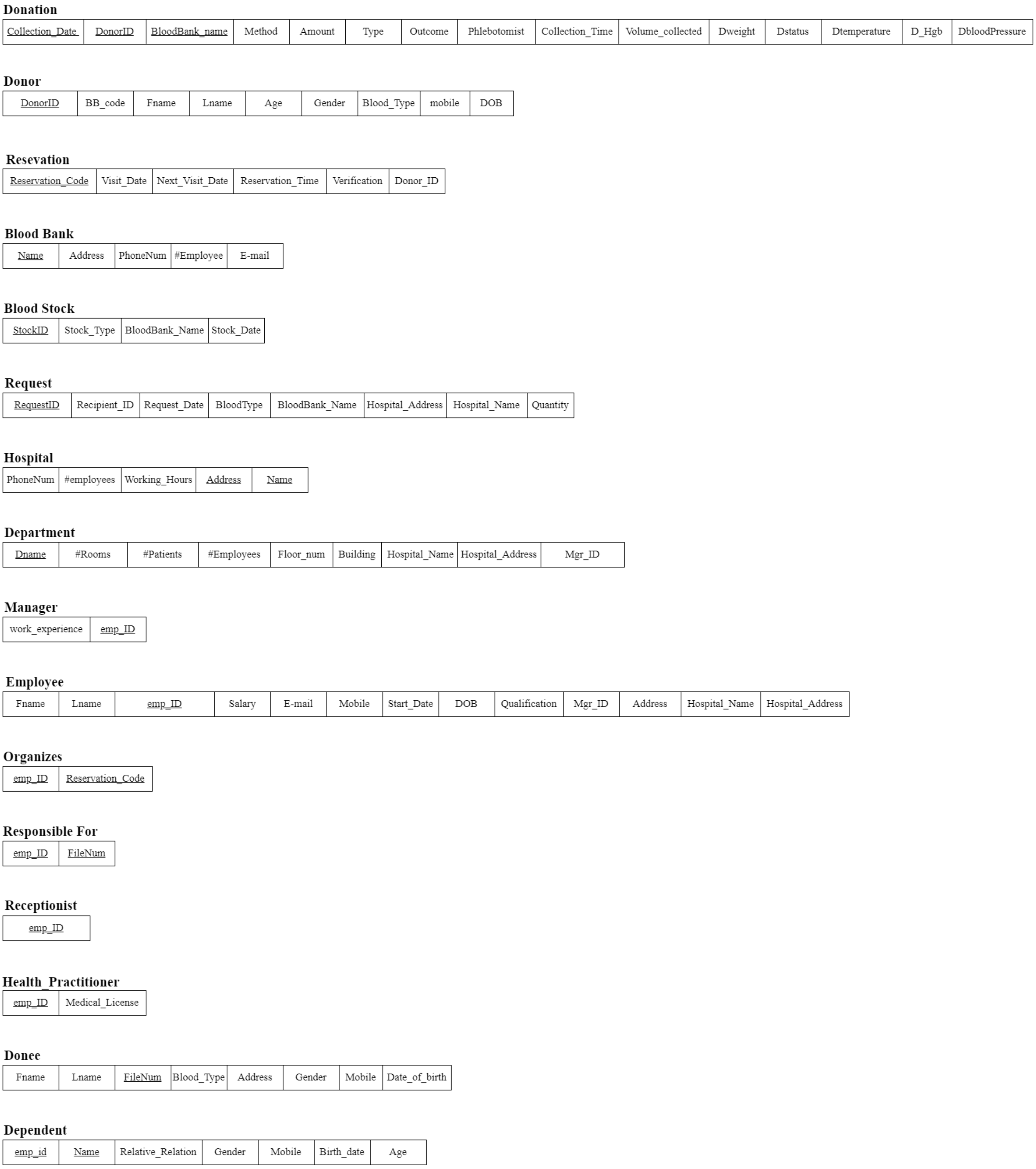
3.8 Schema Diagram



4 Normalization

4.1 First Normal Form

The First Normal Form (1NF) is applied on relations that does not contain multivalued, grouping attributes, and nested relations. All relations (tables) in our relational schema are already in 1NF because all attributes satisfy 1NF rules, simple, and depend on the key.



Donation

Collection_Date	DonorID	BloodBank_name	Method	Amount	Type	Outcome	Phlebotomist	Collection_Time	Volume_collected	Dweight	Dstatus	Dtemperature	D_Hgb	DbloodPressure

Donor

DonorID	BB_code	Fname	Lname	Age	Gender	Blood_Type	mobile	DOB

Resevation

Reservation_Code	Visit_Date	Next_Visit_Date	Reservation_Time	Verification	Donor_ID

Blood Bank

Name	Address	PhoneNum	#Employee	E-mail

Blood Stock

StockID	Stock_Type	BloodBank_Name	Stock_Date

Request

RequestID	Recipient_ID	Request_Date	BloodType	BloodBank_Name	Hospital_Address	Hospital_Name	Quantity

Hospital

PhoneNum	#employees	Working_Hours	Address	Name

Department

Dname	#Rooms	#Patients	#Employees	Floor_num	Building	Hospital_Name	Hospital_Address	Mgr_ID

Manager

work_experience	emp_ID

Employee

Fname	Lname	emp_ID	Salary	E-mail	Mobile	Start_Date	DOB	Qualification	Mgr_ID	Address	Hospital_Name	Hospital_Address

Organizes

emp_ID	Reservation_Code

Responsible For

emp_ID	FileNum

Receptionist

emp_ID

Health_Practitioner

emp_ID	Medical_License

Donee

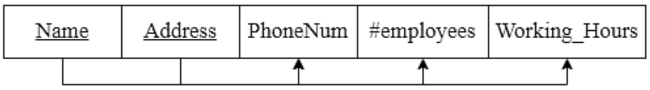
Fname	Lname	FileNum	Blood_Type	Address	Gender	Mobile	Date_of_birth

Dependent

emp_id	Name	Relative_Relation	Gender	Mobile	Birth_date	Age

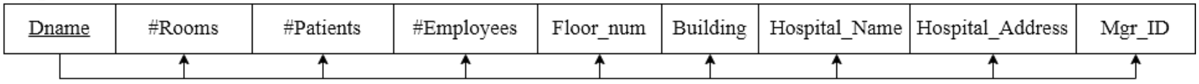
4.2 Second Normal Form

Hospital



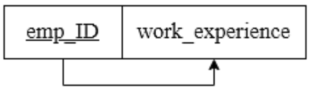
{Address, Name} → PhoneNum, #employee, and Working_Hours are full Functional Dependent (FD) since neither Address→ PhoneNum, #employee, and Working_Hours nor Name→ PhoneNum, #employee, and Working_Hours hold.

Department



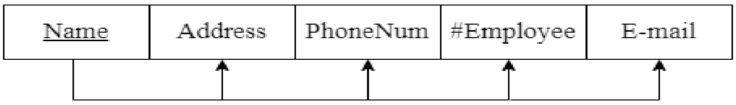
{Dname} → #Rooms, #Patients, #Employees, Floor_num, Building, Hospital_Name, Hospital_Address, and Mgr_ID. All non-primary attributes are fully dependent on the whole key, so there are no partial dependencies. Hence, they are FD.

Manager



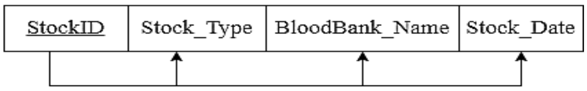
{emp_ID} → work_experience is a full FD since it depends on the key.

Blood Bank

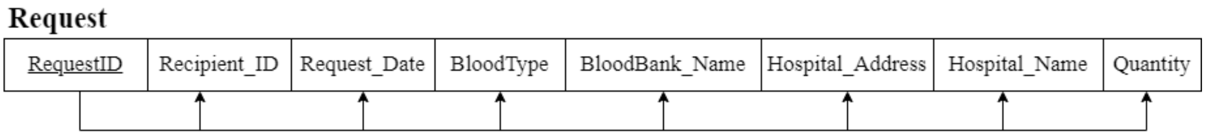


{Name}→ Address, PhoneNum, NumberOf_Employee, and E-mail. All non-primary attributes are fully dependent on the whole key, so there are no partial dependencies. Hence, they are full FD.

Blood Stock



{StockID}→ Stock_Type, BloodBankName, and Stock_Date. All non-primary attributes are fully dependent on the whole key, so there are no partial dependencies. Hence, they are full FD.

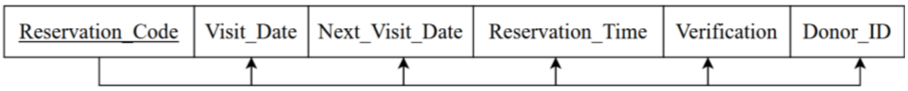


Request

RequestID	Recipient_ID	Request_Date	BloodType	BloodBank_Name	Hospital_Address	Hospital_Name	Quantity

{RequestID}→ RecipientID, Request_Date, Blood_Type, BloodBank_Name, Hospital_Address, Hospital_Name, and Quantity. All non-primary attributes are fully dependent on the whole key, so there are no partial dependencies. Hence, they are full FD.

Reservation

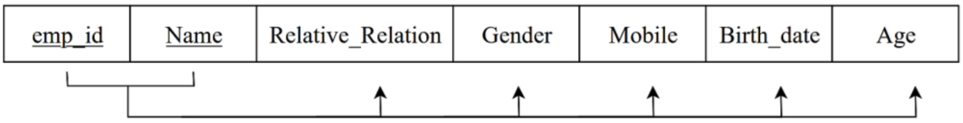


Reservation_Code	Visit_Date	Next_Visit_Date	Reservation_Time	Verification	Donor_ID

{Resevation_Code}→Visit_Date, NextVisit_Date, Reservation_Time, Verification, Donor_ID.

All non-prime attributes are fully dependent on the whole key, so there are no partial dependencies. Hence, they are full FD.

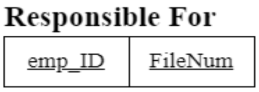
Dependent



emp_id	Name	Relative_Relation	Gender	Mobile	Birth_date	Age

{emp_id, Name}→ Relative_Relation, Gender, Phone, DOB, Age.

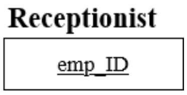
All non-primry attributes are fully dependent on the whole key, so there are no partial dependencies. Hence, they are full FD.



Responsible For

emp_ID	FileNum

The relation has no non-primary keys to be checked, so it's in 2NF since there are no violations.

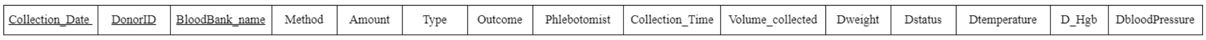


Receptionist

emp_ID

The relation has no non-primary keys to be checked, so it's in 2NF since there are no violations.

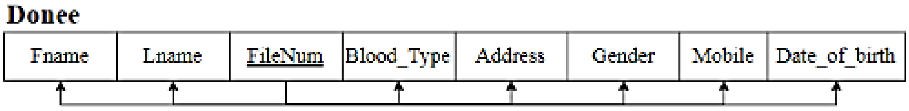
Donation



Collection_Date	DonorID	BloodBank_name	Method	Amount	Type	Outcome	Phlebotomist	Collection_Time	Volume_collected	Dweight	Dstatus	Dtemperature	D_Hgb	DbloodPressure

{Collection_Date, DonorID, BloodBank_name}→Collection_Time, Method, Amount, Type, Outcome, Phlebotomist, Volume_Collected, Dweight, Dstatus, Dtemperature, D_Hgb, and DbloodPressure. All non-primary attributes are fully dependent on the whole key, so there are no partial dependencies. Hence, they are full FD.

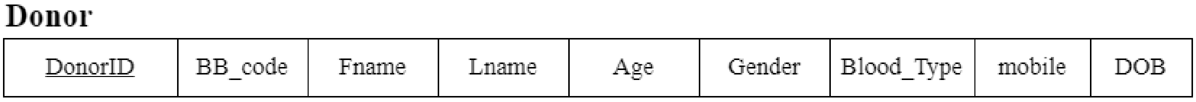
Donee



Fname	Lname	FileNum	Blood_Type	Address	Gender	Mobile	Date_of_birth

{FileNum}→Fname, Lname, Gender, Blood_Type, Mobile, Address, and Date_of_Birth. All non-primary attributes are fully dependent on the whole key, so there are no partial dependencies. Hence, they are full FD.

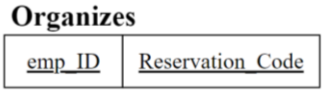
Donor



DonorID	BB_code	Fname	Lname	Age	Gender	Blood_Type	mobile	DOB

{DonorID}→BBcode, Fname, Lname, Age, Gender, Blood_Type, Mobile, and DOB. All non-primary attributes are fully dependent on the whole key, so there are no partial dependencies. Hence, they are full FD.

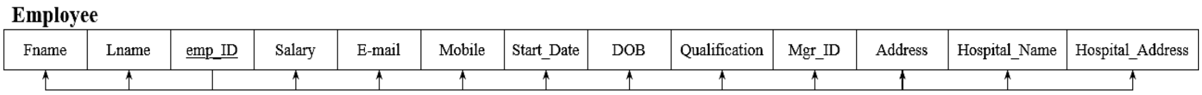
Organizes



emp_ID	Reservation_Code

The relation has no non-primary keys to be checked, so it's in 2NF since there are no violations.

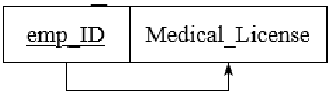
Employee



Fname	Lname	emp_ID	Salary	E-mail	Mobile	Start_Date	DOB	Qualification	Mgr_ID	Address	Hospital_Name	Hospital_Address

{empID}→ Fname, Lname, Salary, E-mail, Mobile, Start_Date, DOB, Qualification, Mrg_ID, Adress,Hospital_Name, Hospital_Adress All non-primary attributes are fully dependent on the whole key, so there are no partial dependencies. Hence, they are full FD.

Health_Practitioner



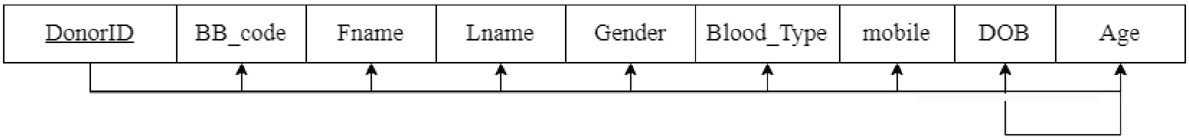
emp_ID	Medical_License

{empID}→ Medical_License is a full FD since it depends on the key.

4.3 Third Normal Form

A Third Normal Form (3NF) relation should not contain a non-primary attribute that transitively depends on the primary key of the relation. All attributes should depend only on the primary key directly.

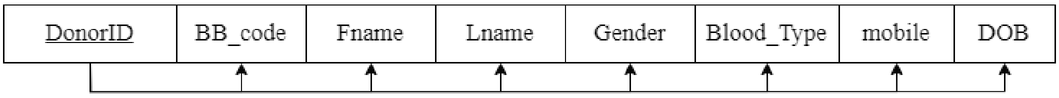
Donor



DonorID	BB_code	Fname	Lname	Gender	Blood_Type	mobile	DOB	Age

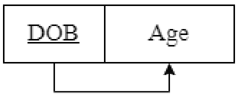
As shown in the relation, Age depends on DOB **(DOB → Age)** which is not a primary key. Therefore, we should normalize it by decomposing Donor into two relations:

Donor



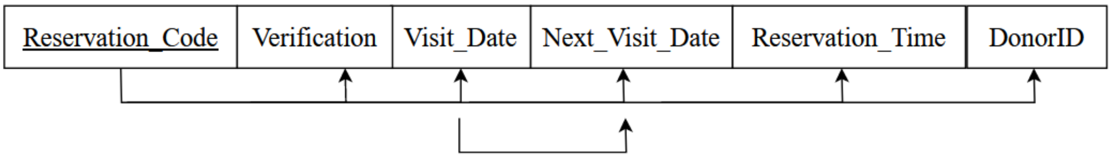
DonorID	BB_code	Fname	Lname	Gender	Blood_Type	mobile	DOB

Donor_age



DOB	Age

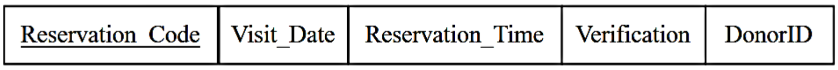
Reservation



Reservation_Code	Verification	Visit_Date	Next_Visit_Date	Reservation_Time	DonorID

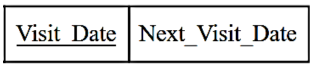
As shown in the relation, Next_Visit_Date depends on Visit_Date **(Visit_Date → Next_Visit_Date)** which is not a primary key. Therefore, we should normalize it by decomposing Reservation into two relations:

Resevation



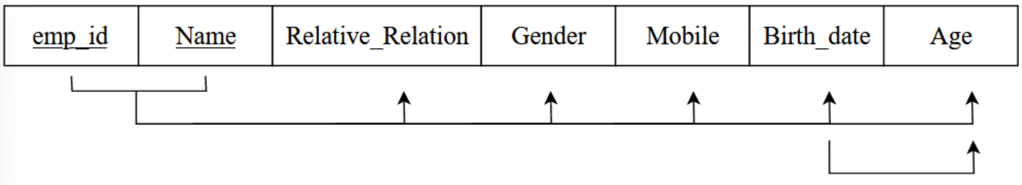
Reservation_Code	Visit_Date	Reservation_Time	Verification	DonorID

Visits



Visit_Date	Next_Visit_Date

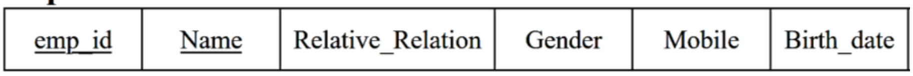
Dependent



emp_id	Name	Relative_Relation	Gender	Mobile	Birth_date	Age

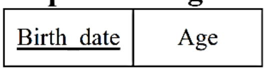
As shown in the relation, Age depends on Birth_Date **(Birth_Date →Age)** which is not a primary key. Therefore, we should normalize it by decomposing Dependent into two relations:

Dependent



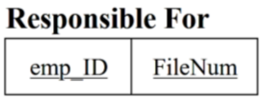
emp_id	Name	Relative_Relation	Gender	Mobile	Birth_date

Dependent Age



Birth_date	Age

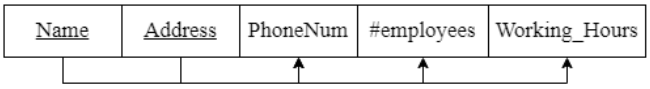
Responsible For



emp_ID	FileNum

The relation has no non-primary keys to be checked (no regular attributes).

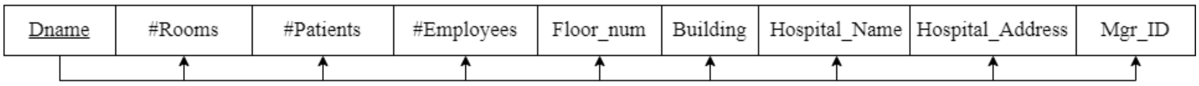
Hospital



Name	Address	PhoneNum	#employees	Working_Hours

Hospital relation is already in 3NF, since all attributes depends on the key, and there are no transitive dependencies.

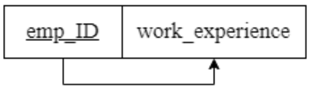
Department



Dname	#Rooms	#Patients	#Employees	Floor_num	Building	Hospital_Name	Hospital_Address	Mgr_ID

Department relation is already in 3NF, since all attributes depends on the key, and there are no transitive dependencies.

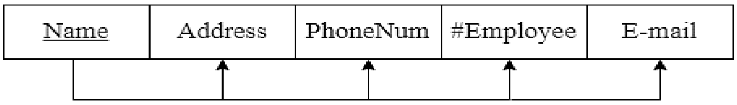
Manager



emp_ID	work_experience

Manager relation is already in 3NF, since all attributes depends on the key, and there is no chance for transitive dependency.

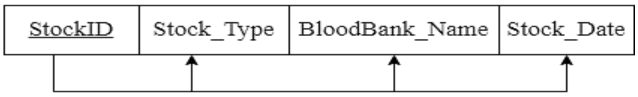
Blood Bank



Name	Address	PhoneNum	#Employee	E-mail

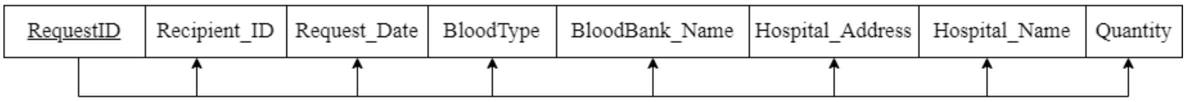
Blood Bank relation is already in 3NF, since all attributes depends on the key, and there are no transitive dependencies.

Blood Stock



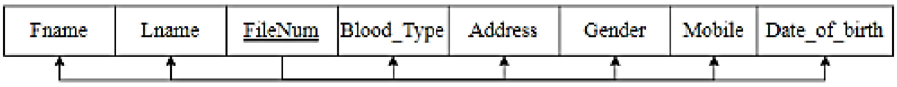
Blood Stock relation is already in 3NF, since all attributes depends on the key, and there are no transitive dependencies.

Request



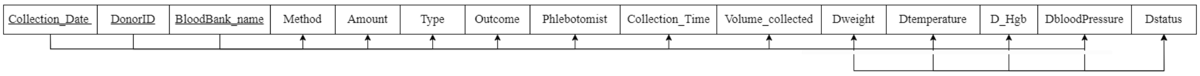
Request relation is already in 3NF, since all attributes depends on the key, and there are no transitive dependencies.

Donee



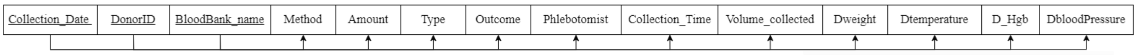
Donee relation is already in 3NF, since all attributes depends on the key, and there are no transitive dependencies.

Donation

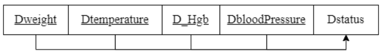


As shown in the relation, Dstatus depends on BDweight, Dtemperature, D_Hgb, and Dbloodpressure **(BDweight, Dtemperature, D_Hgb, Dbloodpressure →Dstatus)** which are not primary keys. Therefore, we should normalize it by decomposing Donation into two relations:

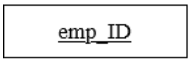
Donation



Donor_status

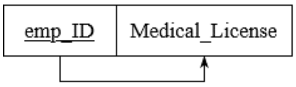


Receptionest



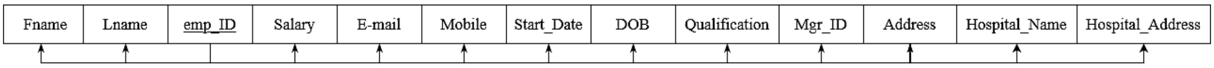
The relation has no non-primary key to be checked.

Health_Practitioner



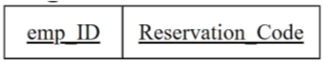
Health_Practitioner relation is already in 3NF, since all attributes depends on the key, and there is no chance for transitive dependency.

Employee



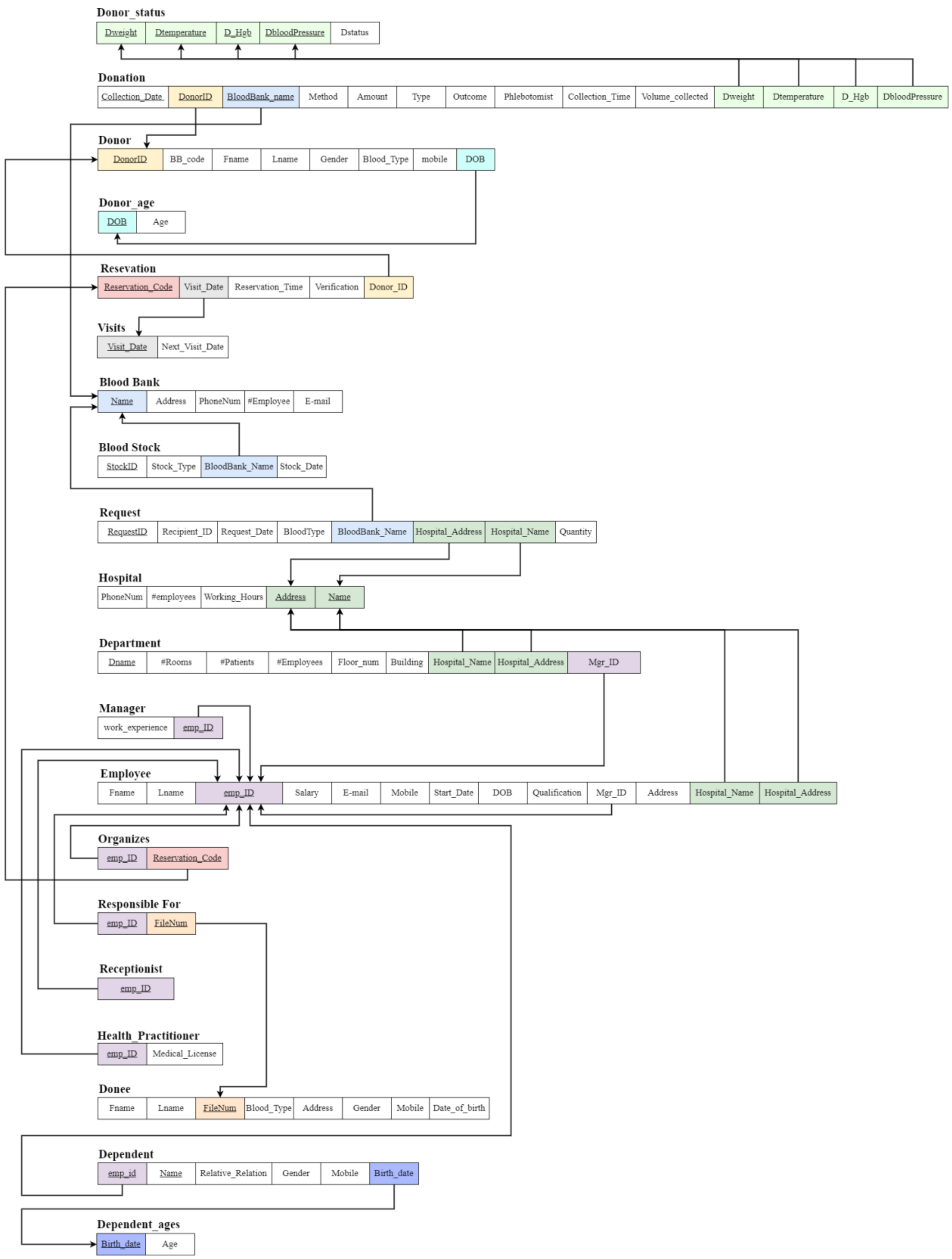
Employee relation is already in 3NF, since all attributes depends on the key, and there is no chance for transitive dependency.

Organizes



Organizes relation is already in 3NF since it’s composed of composite key only and there are no regular attributes. Hence, there is no chance for transitive dependency.

5 Final DB Schema Diagram


Donor_status
Dweight | Dtemperature | D_Hgb | DbloodPressure | Dstatus
Donation
Collection_Date | DonorID | BloodBank_name | Method | Amount | Type | Outcome | Phlebotomist | Collection_Time | Volume_collected | Dweight | Dtemperature | D_Hgb | DbloodPressure
Donor
DonorID | BB_code | Fname | Lname | Gender | Blood_Type | mobile | DOB
Donor_age
DOB | Age
Resevation
Reservation_Code | Visit_Date | Reservation_Time | Verification | Donor_ID
Visits
Visit_Date | Next_Visit_Date
Blood Bank
Name | Address | PhoneNum | #Employee | E-mail
Blood Stock
StockID | Stock_Type | BloodBank_Name | Stock_Date
Request
RequestID | Recipient_ID | Request_Date | BloodType | BloodBank_Name | Hospital_Address | Hospital_Name | Quantity
Hospital
PhoneNum | #employees | Working_Hours | Address | Name
Department
Dname | #Rooms | #Patients | #Employees | Floor_num | Building | Hospital_Name | Hospital_Address | Mgr_ID
Manager
work_experience | emp_ID
Employee
Fname | Lname | emp_ID | Salary | E-mail | Mobile | Start_Date | DOB | Qualification | Mgr_ID | Address | Hospital_Name | Hospital_Address
Organizes
emp_ID | Reservation_Code
Responsible For
emp_ID | FileNum
Receptionist
emp_ID
Health_Practitioner
emp_ID | Medical_License
Donee
Fname | Lname | FileNum | Blood_Type | Address | Gender | Mobile | Date_of_birth
Dependent
emp_id | Name | Relative_Relation | Gender | Mobile | Birth_date
Dependent_ages
Birth_date | Age

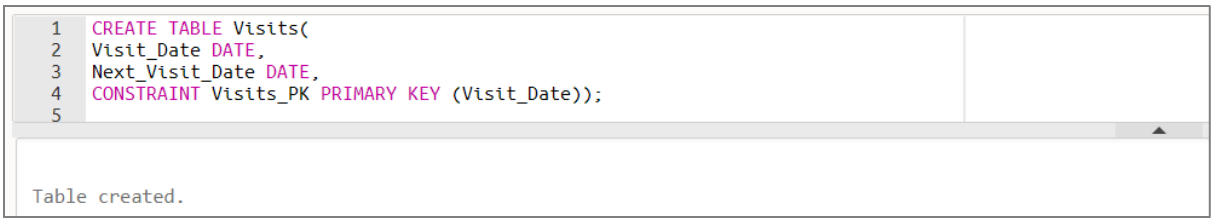
PART III: IMPLEMENTATION

6 Table Creation Script

Show the SQL script of creating all your tables and a screenshot of each script.

6.1 < Visits > TABLE

```
CREATE TABLE Visits(
Visit_Date DATE,
Next_Visit_Date DATE,
CONSTRAINT Visits_PK PRIMARY KEY (Visit_Date));
```

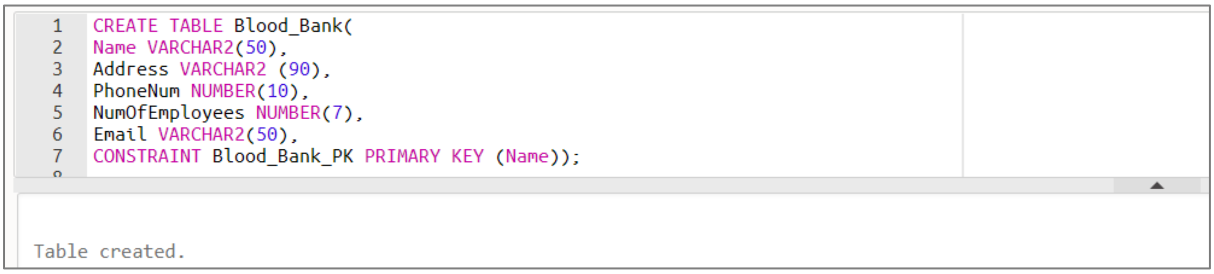


```
CREATE TABLE Visits(
Visit_Date DATE,
Next_Visit_Date DATE,
CONSTRAINT Visits_PK PRIMARY KEY (Visit_Date));
```

```
Table created.
```

6.2 < Blood_Bank > TABLE

```
CREATE TABLE Blood_Bank(
Name VARCHAR2(50),
Address VARCHAR2 (90),
PhoneNum NUMBER(10),
NumOfEmployees NUMBER(7),
Email VARCHAR2(50),
CONSTRAINT Blood_Bank_PK PRIMARY KEY (Name));
```

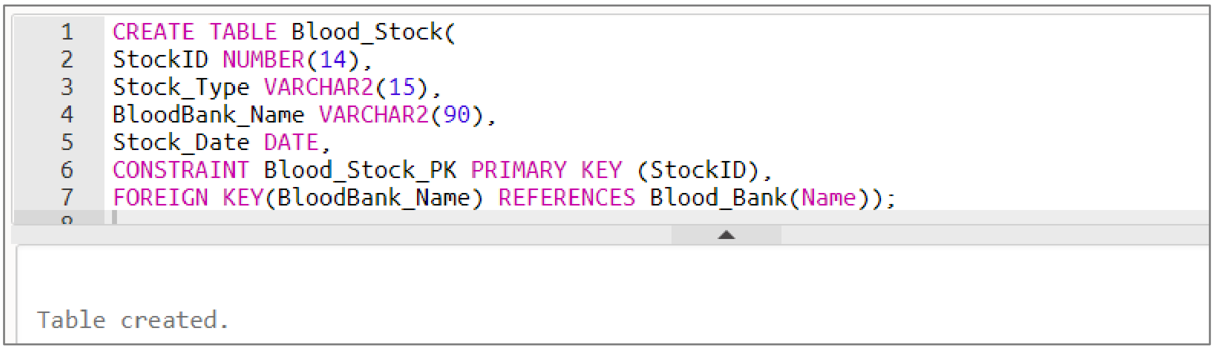


```
CREATE TABLE Blood_Bank(
Name VARCHAR2(50),
Address VARCHAR2 (90),
PhoneNum NUMBER(10),
NumOfEmployees NUMBER(7),
Email VARCHAR2(50),
CONSTRAINT Blood_Bank_PK PRIMARY KEY (Name));
```

```
Table created.
```

6.3 < Blood_Stock > TABLE

```
CREATE TABLE Blood_Stock(
StockID NUMBER(14),
Stock_Type VARCHAR2(15),
BloodBank_Name VARCHAR2(90),
Stock_Date DATE,
CONSTRAINT Blood_Stock_PK PRIMARY KEY (StockID),
FOREIGN KEY(BloodBank_Name) REFERENCES Blood_Bank(Name));
```

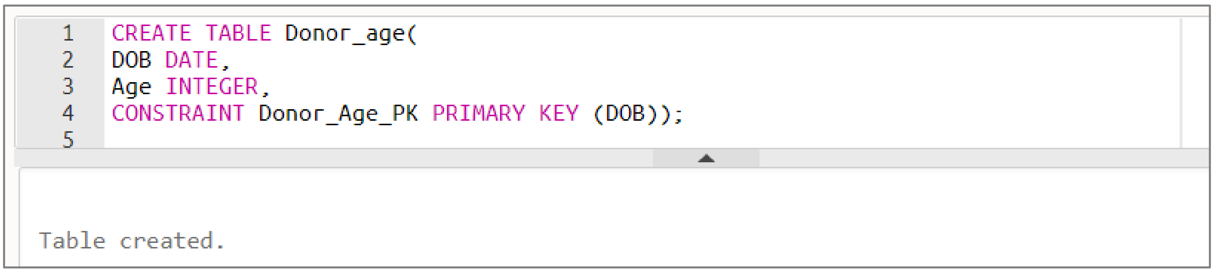


```
CREATE TABLE Blood_Stock(
StockID NUMBER(14),
Stock_Type VARCHAR2(15),
BloodBank_Name VARCHAR2(90),
Stock_Date DATE,
CONSTRAINT Blood_Stock_PK PRIMARY KEY (StockID),
FOREIGN KEY(BloodBank_Name) REFERENCES Blood_Bank(Name));
```

```
Table created.
```

6.4 < Donor_age > TABLE

```
CREATE TABLE Donor_age(
DOB DATE,
Age INTEGER,
CONSTRAINT Donor_Age_PK PRIMARY KEY (DOB));
```

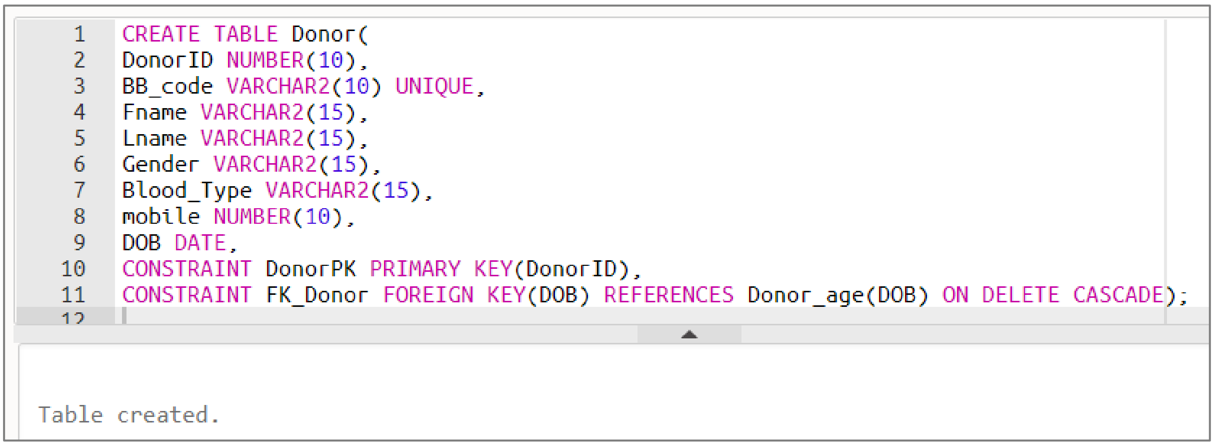


```
CREATE TABLE Donor_age(
DOB DATE,
Age INTEGER,
CONSTRAINT Donor_Age_PK PRIMARY KEY (DOB));
```

```
Table created.
```

6.5 < Donor > TABLE

```
CREATE TABLE Donor(
DonorID NUMBER(10),
BB_code VARCHAR2(10) UNIQUE,
Fname VARCHAR2(15),
Lname VARCHAR2(15),
Gender VARCHAR2(15),
Blood_Type VARCHAR2(15),
mobile NUMBER(10),
DOB DATE,
CONSTRAINT DonorPK PRIMARY KEY(DonorID),
CONSTRAINT FK_Donor FOREIGN KEY(DOB) REFERENCES Donor_age(DOB)
ON DELETE CASCADE);
```

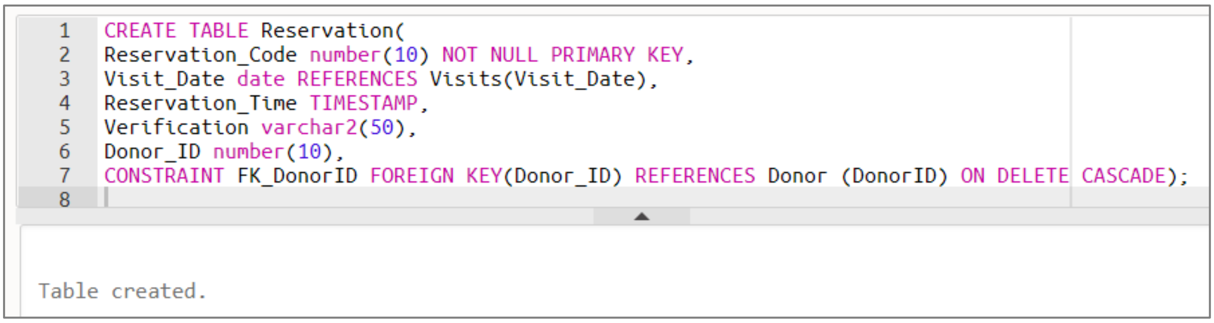


```
CREATE TABLE Donor(
DonorID NUMBER(10),
BB_code VARCHAR2(10) UNIQUE,
Fname VARCHAR2(15),
Lname VARCHAR2(15),
Gender VARCHAR2(15),
Blood_Type VARCHAR2(15),
mobile NUMBER(10),
DOB DATE,
CONSTRAINT DonorPK PRIMARY KEY(DonorID),
CONSTRAINT FK_Donor FOREIGN KEY(DOB) REFERENCES Donor_age(DOB) ON DELETE CASCADE);

Table created.
```

6.6 < Reservation > TABLE

```
CREATE TABLE Reservation(
Reservation_Code number(10) NOT NULL PRIMARY KEY,
Visit_Date date REFERENCES Visits(Visit_Date),
Reservation_Time TIMESTAMP,
Verification varchar2(50),
Donor_ID number(10),
CONSTRAINT FK_DonorID FOREIGN KEY(Donor_ID) REFERENCES Donor
(DonorID) ON DELETE CASCADE);
```



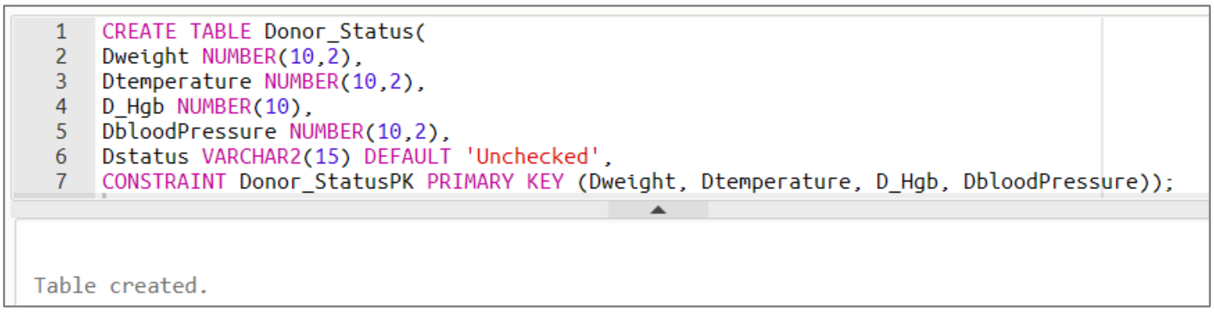
```
CREATE TABLE Reservation(
Reservation_Code number(10) NOT NULL PRIMARY KEY,
Visit_Date date REFERENCES Visits(Visit_Date),
Reservation_Time TIMESTAMP,
Verification varchar2(50),
Donor_ID number(10),
CONSTRAINT FK_DonorID FOREIGN KEY(Donor_ID) REFERENCES Donor (DonorID) ON DELETE CASCADE);

```

```
Table created.
```

6.7 < Donor_Status > TABLE

```
CREATE TABLE Donor_Status(
Dweight NUMBER(10,2),
Dtemperature NUMBER(10,2),
D_Hgb NUMBER(10),
DbloodPressure NUMBER(10,2),
Dstatus VARCHAR2(15) DEFAULT 'Unchecked',
CONSTRAINT Donor_StatusPK PRIMARY KEY (Dweight, Dtemperature,
D_Hgb, DbloodPressure));
```

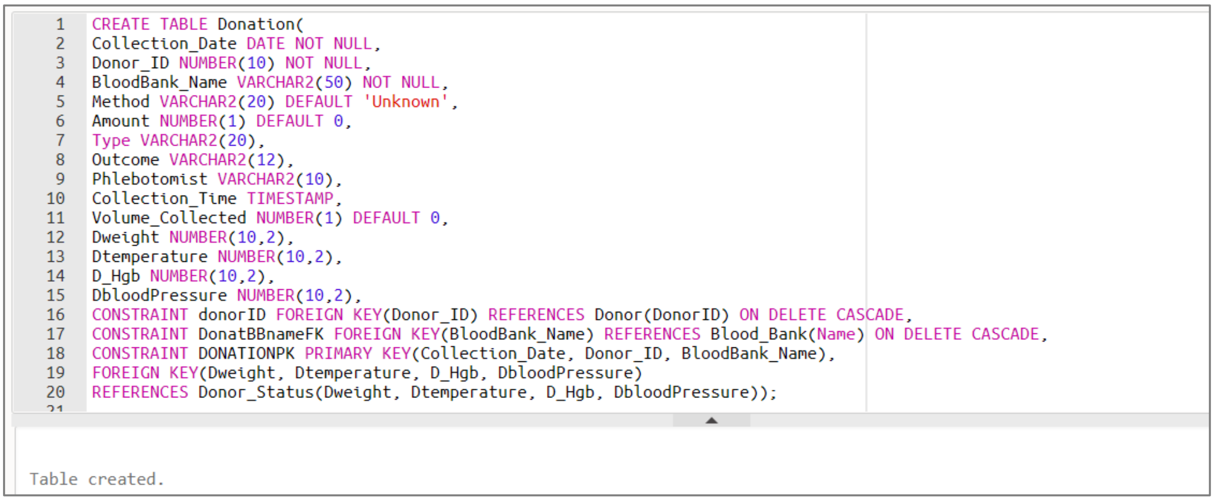


```
CREATE TABLE Donor_Status(
Dweight NUMBER(10,2),
Dtemperature NUMBER(10,2),
D_Hgb NUMBER(10),
DbloodPressure NUMBER(10,2),
Dstatus VARCHAR2(15) DEFAULT 'Unchecked',
CONSTRAINT Donor_StatusPK PRIMARY KEY (Dweight, Dtemperature, D_Hgb, DbloodPressure));
```

```
Table created.
```

6.8 < Donation > TABLE

```
CREATE TABLE Donation(
Collection_Date DATE NOT NULL,
Donor_ID NUMBER(10) NOT NULL,
BloodBank_Name VARCHAR2(50) NOT NULL,
Method VARCHAR2(20) DEFAULT 'Unknown',
Amount NUMBER(1) DEFAULT 0,
Type VARCHAR2(20),
Outcome VARCHAR2(12),
Phlebotomist VARCHAR2(10),
Collection_Time TIMESTAMP,
Volume_Collected NUMBER(1) DEFAULT 0,
Dweight NUMBER(10,2),
Dtemperature NUMBER(10,2),
D_Hgb NUMBER(10,2),
DbloodPressure NUMBER(10,2),
CONSTRAINT donorID FOREIGN KEY(Donor_ID) REFERENCES
Donor(DonorID) ON DELETE CASCADE,
CONSTRAINT DonatBBnameFK FOREIGN KEY(BloodBank_Name)
REFERENCES Blood_Bank(Name) ON DELETE CASCADE,
CONSTRAINT DONATIONPK PRIMARY KEY(Collection_Date, Donor_ID,
BloodBank_Name),
FOREIGN KEY(Dweight, Dtemperature, D_Hgb, DbloodPressure)
REFERENCES Donor_Status(Dweight, Dtemperature, D_Hgb,
DbloodPressure));
```

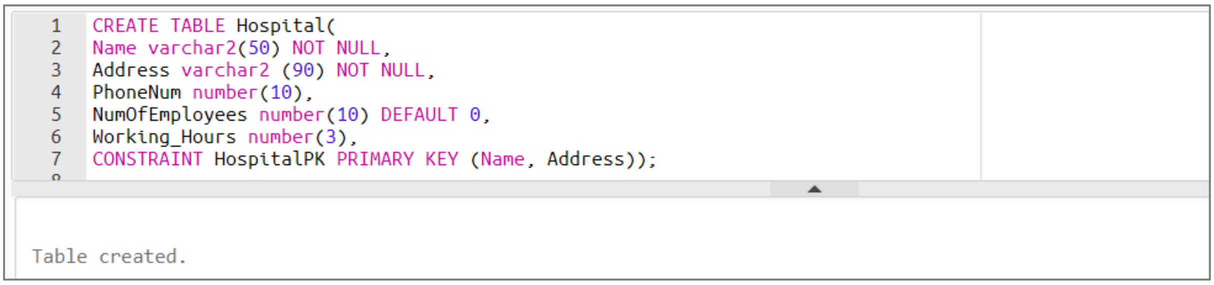


```
CREATE TABLE Donation(
Collection_Date DATE NOT NULL,
Donor_ID NUMBER(10) NOT NULL,
BloodBank_Name VARCHAR2(50) NOT NULL,
Method VARCHAR2(20) DEFAULT 'Unknown',
Amount NUMBER(1) DEFAULT 0,
Type VARCHAR2(20),
Outcome VARCHAR2(12),
Phlebotomist VARCHAR2(10),
Collection_Time TIMESTAMP,
Volume_Collected NUMBER(1) DEFAULT 0,
Dweight NUMBER(10,2),
Dtemperature NUMBER(10,2),
D_Hgb NUMBER(10,2),
DbloodPressure NUMBER(10,2),
CONSTRAINT donorID FOREIGN KEY(Donor_ID) REFERENCES Donor(DonorID) ON DELETE CASCADE,
CONSTRAINT DonatBBnameFK FOREIGN KEY(BloodBank_Name) REFERENCES Blood_Bank(Name) ON DELETE CASCADE,
CONSTRAINT DONATIONPK PRIMARY KEY(Collection_Date, Donor_ID, BloodBank_Name),
FOREIGN KEY(Dweight, Dtemperature, D_Hgb, DbloodPressure)
REFERENCES Donor_Status(Dweight, Dtemperature, D_Hgb, DbloodPressure));
```

```
Table created.
```

6.9 < Hospital > TABLE

```
CREATE TABLE Hospital(
Name varchar2(50) NOT NULL,
Address varchar2 (90) NOT NULL,
PhoneNum number(10),
NumOfEmployees number(10) DEFAULT 0,
Working_Hours number(3),
CONSTRAINT HospitalPK PRIMARY KEY (Name, Address));
```

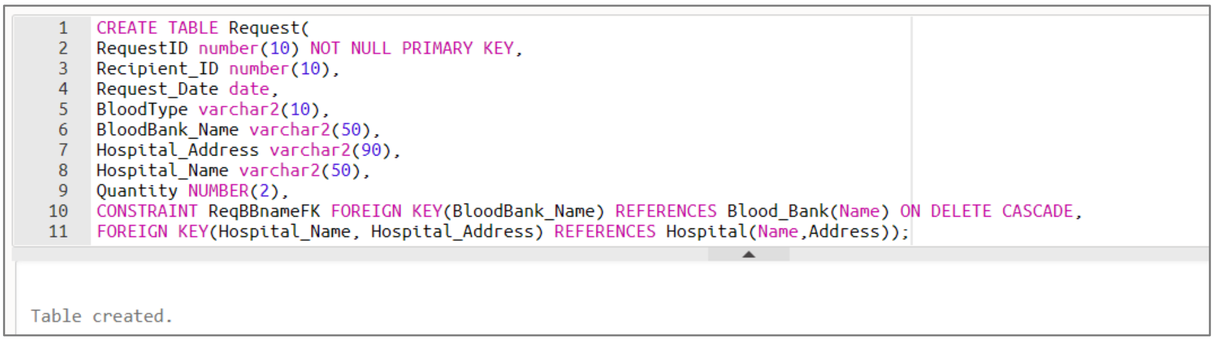


```
CREATE TABLE Hospital(
Name varchar2(50) NOT NULL,
Address varchar2 (90) NOT NULL,
PhoneNum number(10),
NumOfEmployees number(10) DEFAULT 0,
Working_Hours number(3),
CONSTRAINT HospitalPK PRIMARY KEY (Name, Address));
```

```
Table created.
```

6.10 < Request > TABLE

```
CREATE TABLE Request(
RequestID number(10) NOT NULL PRIMARY KEY,
Recipient_ID number(10),
Request_Date date,
BloodType varchar2(10),
BloodBank_Name varchar2(50),
Hospital_Address varchar2(90),
Hospital_Name varchar2(50),
Quantity NUMBER(2),
CONSTRAINT ReqBBnameFK FOREIGN KEY(BloodBank_Name) REFERENCES
Blood_Bank(Name) ON DELETE CASCADE,
FOREIGN KEY(Hospital_Name, Hospital_Address) REFERENCES
Hospital(Name,Address));
```

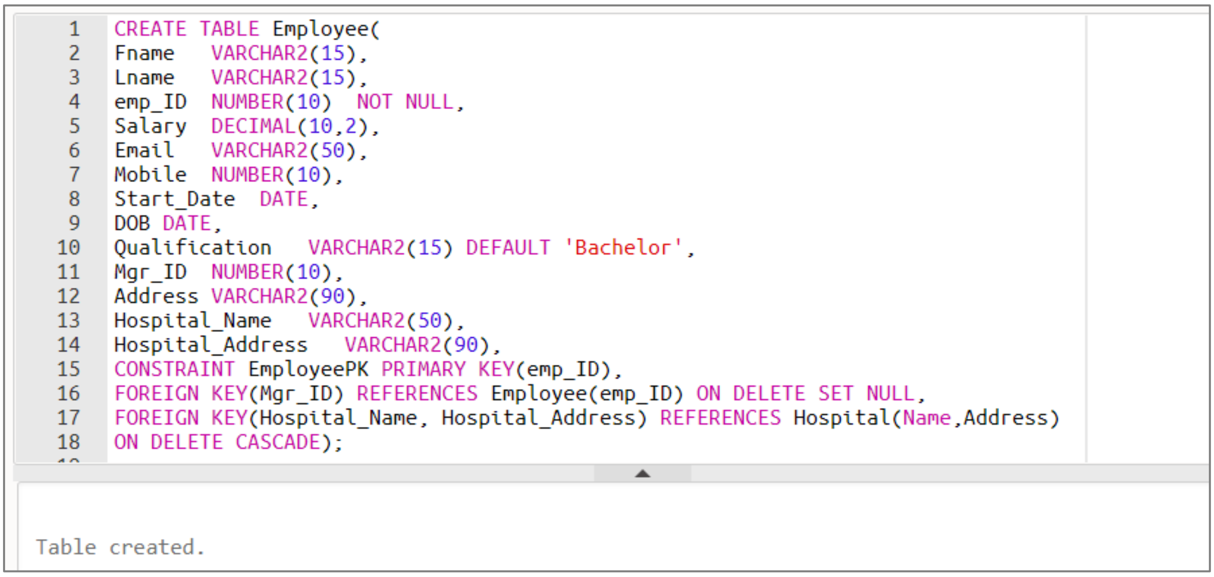


```
CREATE TABLE Request(
RequestID number(10) NOT NULL PRIMARY KEY,
Recipient_ID number(10),
Request_Date date,
BloodType varchar2(10),
BloodBank_Name varchar2(50),
Hospital_Address varchar2(90),
Hospital_Name varchar2(50),
Quantity NUMBER(2),
CONSTRAINT ReqBBnameFK FOREIGN KEY(BloodBank_Name) REFERENCES Blood_Bank(Name) ON DELETE CASCADE,
FOREIGN KEY(Hospital_Name, Hospital_Address) REFERENCES Hospital(Name,Address));
```

```
Table created.
```

6.11 < Employee > TABLE

```
CREATE TABLE Employee(
Fname   VARCHAR2(15),
Lname   VARCHAR2(15),
emp_ID  NUMBER(10)  NOT NULL,
Salary  DECIMAL(10,2),
Email   VARCHAR2(50),
Mobile  NUMBER(10),
Start_Date  DATE,
DOB DATE,
Qualification   VARCHAR2(15) DEFAULT 'Bachelor',
Mgr_ID  NUMBER(10),
Address VARCHAR2(90),
Hospital_Name   VARCHAR2(50),
Hospital_Address   VARCHAR2(90),
CONSTRAINT EmployeePK PRIMARY KEY(emp_ID),
FOREIGN KEY(Mgr_ID) REFERENCES Employee(emp_ID) ON DELETE SET
NULL,
FOREIGN KEY(Hospital_Name, Hospital_Address) REFERENCES
Hospital(Name,Address) ON DELETE CASCADE
);
```

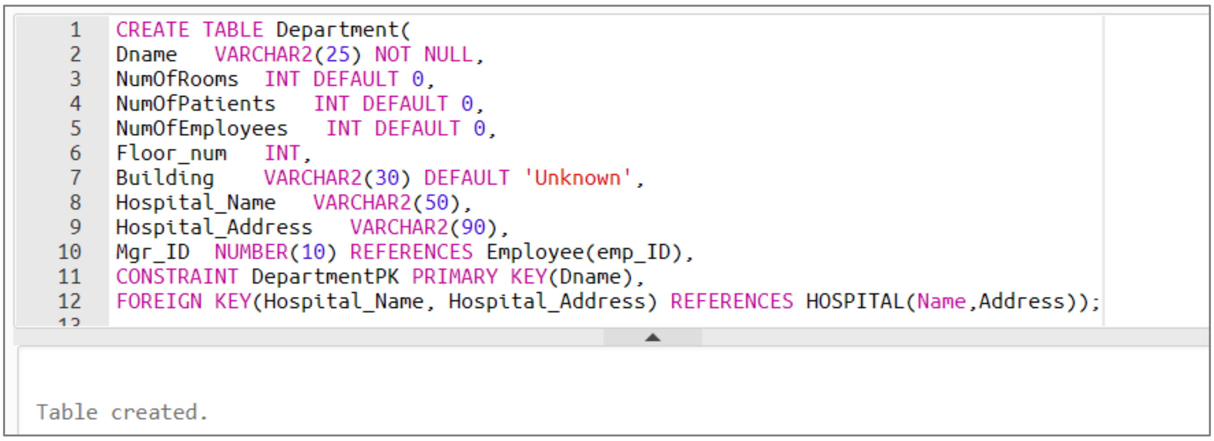


```
CREATE TABLE Employee(
Fname   VARCHAR2(15),
Lname   VARCHAR2(15),
emp_ID  NUMBER(10)  NOT NULL,
Salary  DECIMAL(10,2),
Email   VARCHAR2(50),
Mobile  NUMBER(10),
Start_Date  DATE,
DOB DATE,
Qualification   VARCHAR2(15) DEFAULT 'Bachelor',
Mgr_ID  NUMBER(10),
Address VARCHAR2(90),
Hospital_Name   VARCHAR2(50),
Hospital_Address   VARCHAR2(90),
CONSTRAINT EmployeePK PRIMARY KEY(emp_ID),
FOREIGN KEY(Mgr_ID) REFERENCES Employee(emp_ID) ON DELETE SET NULL,
FOREIGN KEY(Hospital_Name, Hospital_Address) REFERENCES Hospital(Name,Address)
ON DELETE CASCADE);
```

```
Table created.
```

6.12 < Department > TABLE

```
CREATE TABLE Department(
Dname   VARCHAR2(25) NOT NULL,
NumOfRooms  INT DEFAULT 0,
NumOfPatients   INT DEFAULT 0,
NumOfEmployees   INT DEFAULT 0,
Floor_num   INT,
Building    VARCHAR2(30) DEFAULT 'Unknown',
Hospital_Name   VARCHAR2(50),
Hospital_Address   VARCHAR2(90),
Mgr_ID  NUMBER(10) REFERENCES Employee(emp_ID),
CONSTRAINT DepartmentPK PRIMARY KEY(Dname),
FOREIGN KEY(Hospital_Name, Hospital_Address) REFERENCES
HOSPITAL(Name,Address));
```

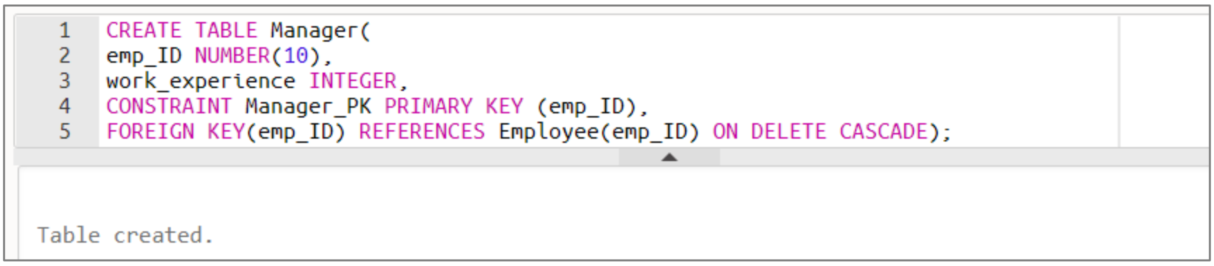


```
CREATE TABLE Department(
Dname   VARCHAR2(25) NOT NULL,
NumOfRooms  INT DEFAULT 0,
NumOfPatients   INT DEFAULT 0,
NumOfEmployees   INT DEFAULT 0,
Floor_num   INT,
Building    VARCHAR2(30) DEFAULT 'Unknown',
Hospital_Name   VARCHAR2(50),
Hospital_Address   VARCHAR2(90),
Mgr_ID  NUMBER(10) REFERENCES Employee(emp_ID),
CONSTRAINT DepartmentPK PRIMARY KEY(Dname),
FOREIGN KEY(Hospital_Name, Hospital_Address) REFERENCES HOSPITAL(Name,Address));
```

```
Table created.
```

6.13 < Manager > TABLE

```
CREATE TABLE Manager(
emp_ID NUMBER(10),
work_experience INTEGER,
CONSTRAINT Manager_PK PRIMARY KEY (emp_ID),
FOREIGN KEY(emp_ID) REFERENCES Employee(emp_ID) ON DELETE
CASCADE);
```

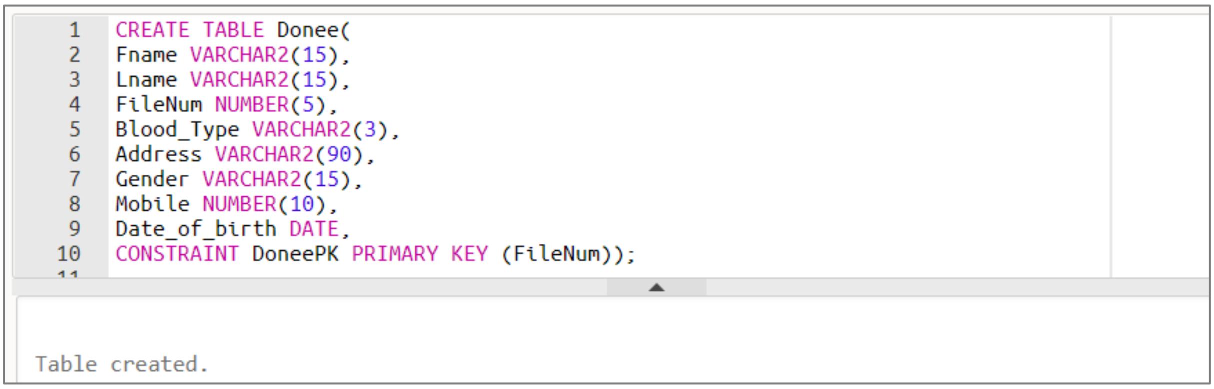


```
CREATE TABLE Manager(
emp_ID NUMBER(10),
work_experience INTEGER,
CONSTRAINT Manager_PK PRIMARY KEY (emp_ID),
FOREIGN KEY(emp_ID) REFERENCES Employee(emp_ID) ON DELETE CASCADE);
```

```
Table created.
```

6.14 < Donee > TABLE

```
CREATE TABLE Donee(
Fname VARCHAR2(15),
Lname VARCHAR2(15),
FileNum NUMBER(5),
Blood_Type VARCHAR2(3),
Address VARCHAR2(90),
Gender VARCHAR2(15),
Mobile NUMBER(10),
Date_of_birth DATE,
CONSTRAINT DoneePK PRIMARY KEY (FileNum));
```

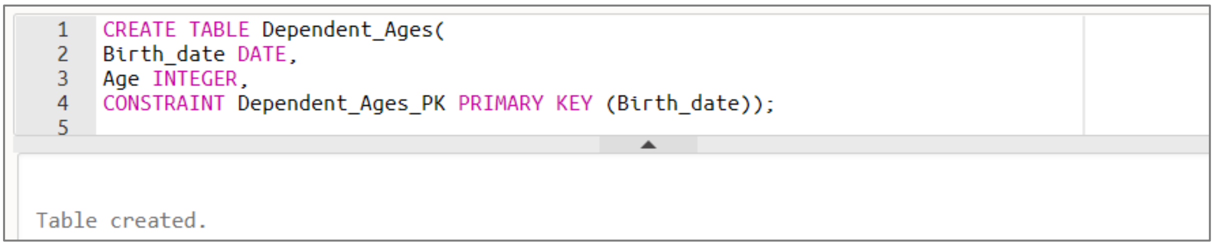


```
CREATE TABLE Donee(
Fname VARCHAR2(15),
Lname VARCHAR2(15),
FileNum NUMBER(5),
Blood_Type VARCHAR2(3),
Address VARCHAR2(90),
Gender VARCHAR2(15),
Mobile NUMBER(10),
Date_of_birth DATE,
CONSTRAINT DoneePK PRIMARY KEY (FileNum));
```

```
Table created.
```

6.15 < Dependent_Ages > TABLE

```
CREATE TABLE Dependent_Ages(
Birth_date DATE,
Age INTEGER,
CONSTRAINT Dependent_Ages_PK PRIMARY KEY (Birth_date));
```

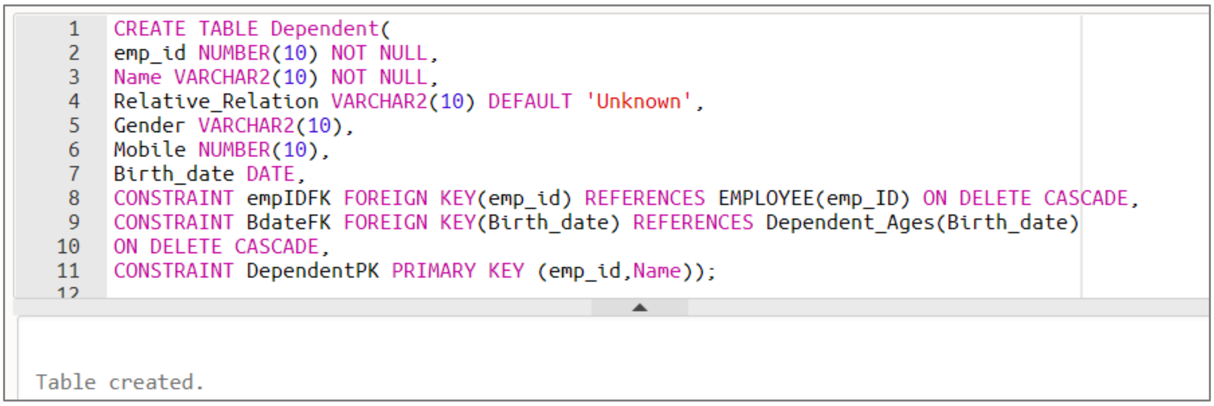


```
CREATE TABLE Dependent_Ages(
Birth_date DATE,
Age INTEGER,
CONSTRAINT Dependent_Ages_PK PRIMARY KEY (Birth_date));
```

```
Table created.
```

6.16 < Dependent > TABLE

```
CREATE TABLE Dependent(
emp_id NUMBER(10) NOT NULL,
Name VARCHAR2(10) NOT NULL,
Relative_Relation VARCHAR2(10) DEFAULT 'Unknown',
Gender VARCHAR2(10),
Mobile NUMBER(10),
Birth_date DATE,
CONSTRAINT empIDFK FOREIGN KEY(emp_id) REFERENCES
EMPLOYEE(emp_ID) ON DELETE CASCADE,
CONSTRAINT BdateFK FOREIGN KEY(Birth_date) REFERENCES
Dependent_Ages(Birth_date) ON DELETE CASCADE,
CONSTRAINT DependentPK PRIMARY KEY (emp_id,Name));
```

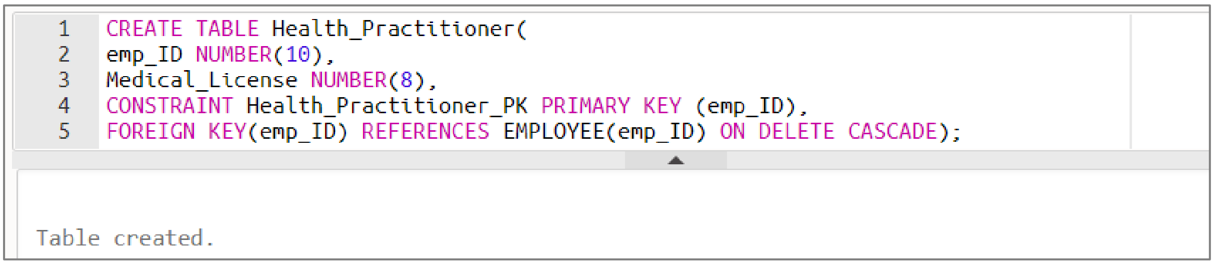


```
CREATE TABLE Dependent(
emp_id NUMBER(10) NOT NULL,
Name VARCHAR2(10) NOT NULL,
Relative_Relation VARCHAR2(10) DEFAULT 'Unknown',
Gender VARCHAR2(10),
Mobile NUMBER(10),
Birth_date DATE,
CONSTRAINT empIDFK FOREIGN KEY(emp_id) REFERENCES EMPLOYEE(emp_ID) ON DELETE CASCADE,
CONSTRAINT BdateFK FOREIGN KEY(Birth_date) REFERENCES Dependent_Ages(Birth_date)
ON DELETE CASCADE,
CONSTRAINT DependentPK PRIMARY KEY (emp_id,Name));
```

```
Table created.
```

6.17 < Health_Practitioner > TABLE

```
CREATE TABLE Health_Practitioner(
emp_ID NUMBER(10),
Medical_License NUMBER(8),
CONSTRAINT Health_Practitioner_PK PRIMARY KEY (emp_ID),
FOREIGN KEY(emp_ID) REFERENCES EMPLOYEE(emp_ID) ON DELETE
CASCADE);
```

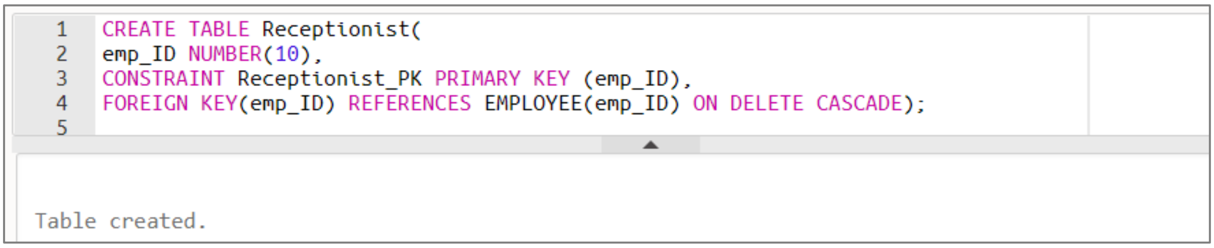


```
CREATE TABLE Health_Practitioner(
emp_ID NUMBER(10),
Medical_License NUMBER(8),
CONSTRAINT Health_Practitioner_PK PRIMARY KEY (emp_ID),
FOREIGN KEY(emp_ID) REFERENCES EMPLOYEE(emp_ID) ON DELETE CASCADE);
```

```
Table created.
```

6.18 < Receptionist > TABLE

```
CREATE TABLE Receptionist(
emp_ID NUMBER(10),
CONSTRAINT Receptionist_PK PRIMARY KEY (emp_ID),
FOREIGN KEY(emp_ID) REFERENCES EMPLOYEE(emp_ID) ON DELETE
CASCADE);
```



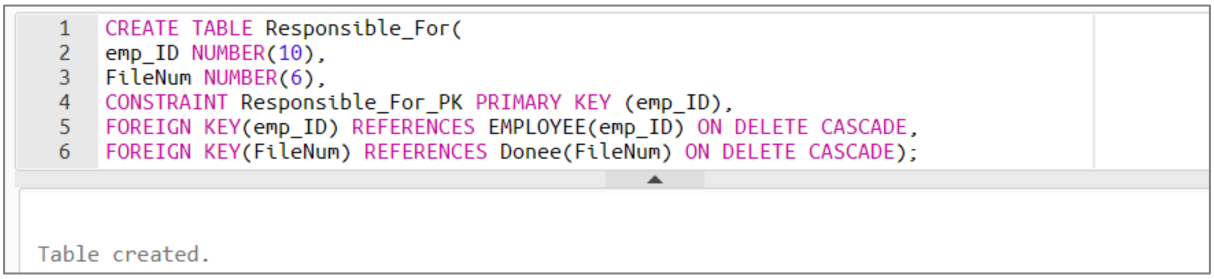
```
CREATE TABLE Receptionist(
emp_ID NUMBER(10),
CONSTRAINT Receptionist_PK PRIMARY KEY (emp_ID),
FOREIGN KEY(emp_ID) REFERENCES EMPLOYEE(emp_ID) ON DELETE CASCADE);

```

```
Table created.
```

6.19 < Responsible_For > TABLE

```
CREATE TABLE Responsible_For(
emp_ID NUMBER(10),
FileNum NUMBER(6),
CONSTRAINT Responsible_For_PK PRIMARY KEY (emp_ID),
FOREIGN KEY(emp_ID) REFERENCES EMPLOYEE(emp_ID) ON DELETE
CASCADE,
FOREIGN KEY(FileNum) REFERENCES Donee(FileNum) ON DELETE
CASCADE);
```

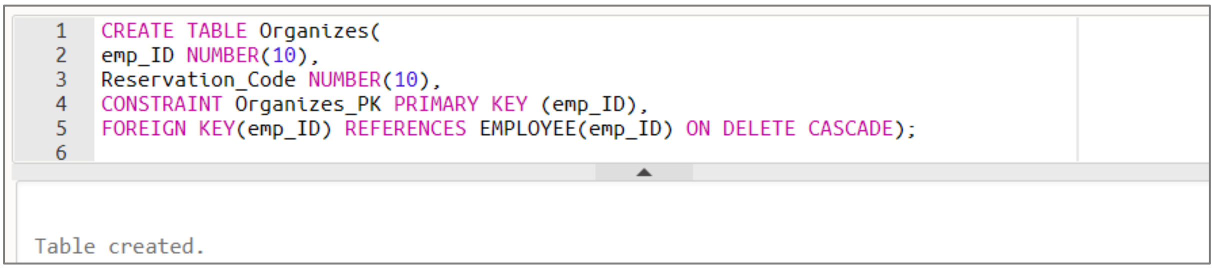


```
CREATE TABLE Responsible_For(
emp_ID NUMBER(10),
FileNum NUMBER(6),
CONSTRAINT Responsible_For_PK PRIMARY KEY (emp_ID),
FOREIGN KEY(emp_ID) REFERENCES EMPLOYEE(emp_ID) ON DELETE CASCADE,
FOREIGN KEY(FileNum) REFERENCES Donee(FileNum) ON DELETE CASCADE);
```

```
Table created.
```

6.20 < Organizes > TABLE

```
CREATE TABLE Organizes(
emp_ID NUMBER(10),
Reservation_Code NUMBER(10),
CONSTRAINT Organizes_PK PRIMARY KEY (emp_ID),
FOREIGN KEY(emp_ID) REFERENCES EMPLOYEE(emp_ID) ON DELETE
CASCADE);
```



```
CREATE TABLE Organizes(
emp_ID NUMBER(10),
Reservation_Code NUMBER(10),
CONSTRAINT Organizes_PK PRIMARY KEY (emp_ID),
FOREIGN KEY(emp_ID) REFERENCES EMPLOYEE(emp_ID) ON DELETE CASCADE);

```

```
Table created.
```

7 Constraints Script

Business Rule	SQL Script	Table
The employee must have a bachelor’s degree.	Qualification VARCHAR2(15) NOT NULL DEFAULT 'Bachelor',	Employee
Each employee has unique ID.	emp_ID NUMBER(10) NOT NULL, CONSTRAINT EmployeePK PRIMARY KEY(emp_ID),	Employee

Must have a verified medical license.	Medical_License VARCHAR2(9) NOT NULL,	Health_Practitioner
A manager must have 10 years of experience in the field.	ALTER TABLE Manager ADD CONSTRAINT ExpYears CHECK (work_experience >= 10);	Manager
Urgent need of blood.	CONSTRAINT Responsible_For_PK PRIMARY KEY (emp_ID), FOREIGN KEY(emp_ID) REFERENCES EMPLOYEE(emp_ID) ON DELETE CASCADE, FOREIGN KEY(FileNum) REFERENCES Donee(FileNum) ON DELETE CASCADE);	Responsible_For
For a doner, the reservation date must be 8 weeks at least after the last donation.	ALTER TABLE Visits ADD CONSTRAINT nextVisitCHK CHECK (Next_Visit_Date >= ADD_MONTHS(Visit_Date, 2));	Visits
Each bag of blood must have Special ID.	StockID NUMBER(14), CONSTRAINT Blood_Stock_PK PRIMARY KEY (StockID),	Blood_Stock
Each Department Should have a manager.	Mgr_ID NUMBER(10) NOT NULL, CONSTRAINT mgrFK FOREIGN KEY(Mgr_ID) REFERENCES Employee(emp_ID),	Department
Age must be greater than 17 and less than 66 years.	ALTER TABLE Donor_age ADD CONSTRAINT DonorAgeCHK CHECK (Age BETWEEN 17 AND 66);	Donor_age
Each donor must have a unique Blood bank ID that starts with BB followed by followed by 6 digits.	BB_code VARCHAR2(8) UNIQUE,	Donor
Weight must be at least 50 kg.	ALTER TABLE Donor_Status ADD CONSTRAINT CHK_Weight CHECK (Dweight >=50);	Donor_Status
Haemoglobin level must be 13-17	ALTER TABLE Donor_Status ADD CONSTRAINT CHK_Hgb CHECK (D_Hgb BETWEEN 13 AND 17);	Donor_Status
Donors’ blood pressure should not be higher than 180/100 mmHg.	ALTER TABLE Donor_Status ADD CONSTRAINT BpressCHK CHECK (DbloodPressure <= 180/100);	Donor_Status
Donors’ body temperature should not be higher than37.5°C.	ALTER TABLE Donor_Status ADD CONSTRAINT TempCHK CHECK (Dtemperature <= 37.5);	Donor_Status
donor status must be valid to be able to donate	ALTER TABLE Donor_Status ADD CONSTRAINT	Donor_Status

	StatusCHK CHECK (Dstatus = 'valid');	
Donor donates to the blood bank.	Collection_Date DATE NOT NULL, CONSTRAINT DONATIONPK PRIMARY KEY(Collection_Date, Donor_ID, BloodBank_Name), CONSTRAINT donorID FOREIGN KEY(Donor_ID) REFERENCES Donor(DonorID) ON DELETE CASCADE, CONSTRAINT DonatBBnameFK FOREIGN KEY(BloodBank_Name) REFERENCES Blood_Bank(Name) ON DELETE CASCADE, FOREIGN KEY(Dweight, Dtemperature, D_Hgb, DbloodPressure) REFERENCES Donor_Status(Dweight, Dtemperature, D_Hgb, DbloodPressure));	Donation
Donor can donate up to two bags in every donation	ALTER TABLE Donation ADD CONSTRAINT AMTLimit CHECK (Amount <= 2);	Donation
Each Blood bag is stored in bloodstock.	StockID NUMBER(14) NOT NULL, CONSTRAINT BloodStock_PK PRIMARY KEY (StockID), FOREIGN KEY(BloodBank_Name) REFERENCES Blood_Bank(Name));	Blood_Stock

8 Queries and Transactions

In the following subsections, write down five different SQL queries which implements five of the intended output of your system (q.v. Section 1.4). They should be the most important queries. Also, show examples of one update and one delete.

8.1 < Display max number of reservations by a donor>

Query in Natural Language (English)

Retrieve the donorID and number of reservations of the donor that has the maximum number of reservations.

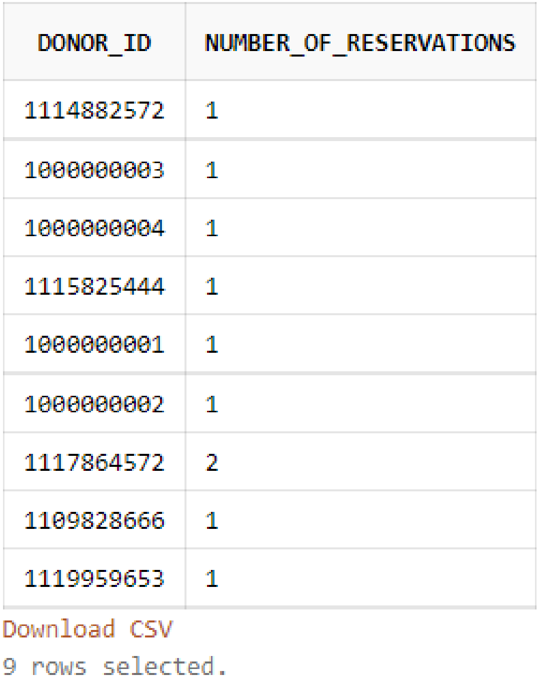
First, we will display a table of each donorID and his number of reservations just to show which one of them has the max number of reservation and make sure our final query is correct. Then, we will display the result of the main query.

SQL Script

```
SELECT Donor_ID, COUNT(Donor_ID) AS Number_of_Reservations
 FROM Reservation
 GROUP BY Donor_ID;
```

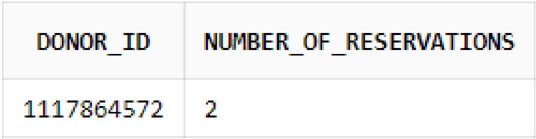
```
SELECT Donor_ID, COUNT(Donor_ID) AS Number_of_Reservations
FROM Reservation
GROUP BY Donor_ID
HAVING COUNT(Donor_ID) = (SELECT MAX(COUNT(Donor_ID))
                           FROM Reservation
                           GROUP BY Donor_ID);
```

Caption of the First Five Rows of the Output



DONOR_ID	NUMBER_OF_RESERVATIONS
1114882572	1
1000000003	1
1000000004	1
1115825444	1
1000000001	1
1000000002	1
1117864572	2
1109828666	1
1119959653	1

Download CSV
9 rows selected.



DONOR_ID	NUMBER_OF_RESERVATIONS
1117864572	2

8.2 < Display donation records with the ability to specify some properties >

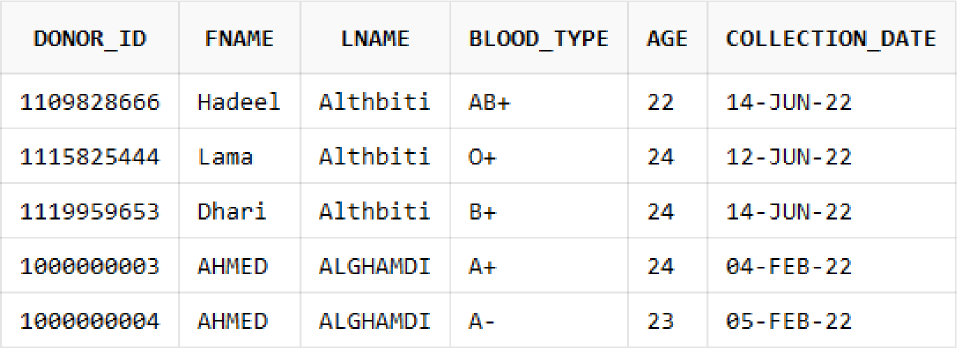
Query in Natural Language (English)

Retrieve all donation records that are placed after a donor’s First name is “ALI” donated, and print donorID, first and last name, blood type, age, and the collection date for all retrieved records.

SQL Script

```
SELECT Donor_ID, Fname, Lname, Blood_Type, Age, Collection_Date
FROM Donation, Donor, Donor_age
WHERE Donor_ID = DonorID AND Donor.DOB = Donor_age.DOB AND
 Collection_Date > ( SELECT Collection_Date
FROM Donation, Donor
WHERE Donation.Donor_ID = Donor.DonorID And Fname = 'ALI');
```

Caption of the First Five Rows of the Output



DONOR_ID	FNAME	LNAME	BLOOD_TYPE	AGE	COLLECTION_DATE
1109828666	Hadeel	Althbiti	AB+	22	14-JUN-22
1115825444	Lama	Althbiti	O+	24	12-JUN-22
1119959653	Dhari	Althbiti	B+	24	14-JUN-22
1000000003	AHMED	ALGHAMDI	A+	24	04-FEB-22
1000000004	AHMED	ALGHAMDI	A-	23	05-FEB-22

8.3 < Display the unavailable blood types >

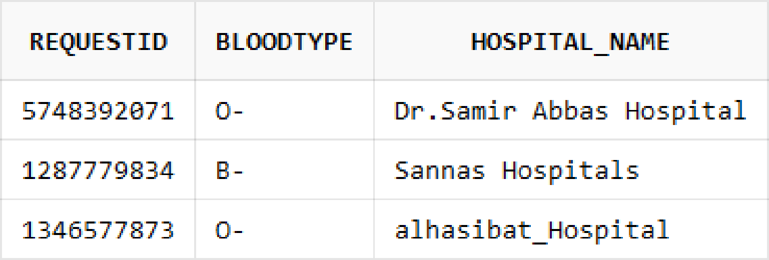
Query in Natural Language (English)

Extract the blood requests where the blood type is not available in any of the blood stocks

SQL Script

```
SELECT RequestID, BloodType, Hospital_Name
FROM Request
WHERE NOT EXISTS(
                SELECT BloodType
                FROM BLood_stock
                WHERE BLood_stock.Stock_Type = Request.BloodType);
```

Caption of the First Five Rows of the Output



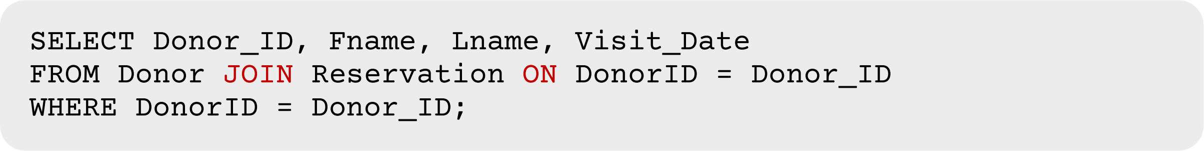
REQUESTID	BLOODTYPE	HOSPITAL_NAME
5748392071	O-	Dr.Samir Abbas Hospital
1287779834	B-	Sannas Hospitals
1346577873	O-	alhasibat_Hospital

8.4 <Display Donor's Information with visit dates >

Query in Natural Language (English)

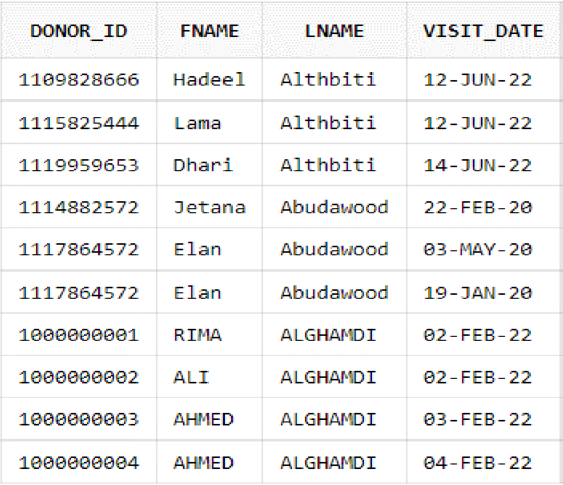
Retrieves all donors who made reservations and display their ID, first name, last name, and the visit date.

SQL Script



```
SELECT Donor_ID, Fname, Lname, Visit_Date
FROM Donor JOIN Reservation ON DonorID = Donor_ID
WHERE DonorID = Donor_ID;
```

Caption of the First Five Rows of the Output



DONOR_ID	FNAME	LNAME	VISIT_DATE
1109828666	Hadeel	Althbiti	12-JUN-22
1115825444	Lama	Althbiti	12-JUN-22
1119959653	Dhari	Althbiti	14-JUN-22
1114882572	Jetana	Abudawood	22-FEB-20
1117864572	Elan	Abudawood	03-MAY-20
1117864572	Elan	Abudawood	19-JAN-20
1000000001	RIMA	ALGHAMDI	02-FEB-22
1000000002	ALI	ALGHAMDI	02-FEB-22
1000000003	AHMED	ALGHAMDI	03-FEB-22
1000000004	AHMED	ALGHAMDI	04-FEB-22

8.5 < Display the verified donors before blood donation >

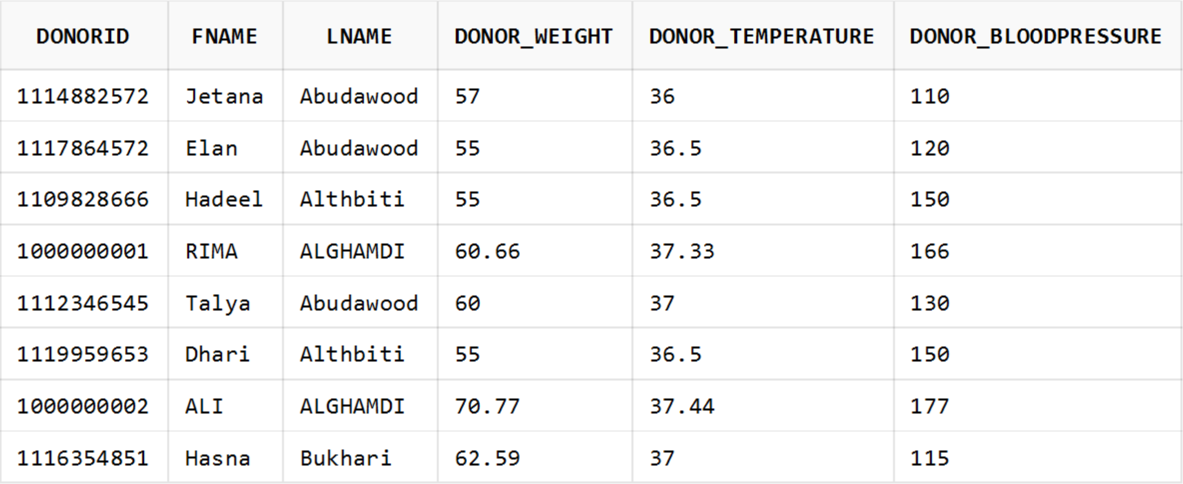
Query in Natural Language (English)

Retrieve donors’ information after verification and before blood transfusion.

SQL Script

```
SELECT DISTINCT DonorID, Fname, Lname, Donation.Dweight AS
DONOR_weight , Donation.Dtemperature AS DONOR_temperature,
Donation.DbloodPressure AS DONOR_bloodPressure
FROM Donation
INNER JOIN Donor ON  Donation.Donor_ID = Donor.DonorID AND
Donation.Dweight >50 AND
Donation.Dtemperature < 37.5 AND
(Donation.D_Hgb BETWEEN 13 AND 17)AND
Donation.DbloodPressure < 180 ;
```

Caption of the First Five Rows of the Output



DONORID	FNAME	LNAME	DONOR_WEIGHT	DONOR_TEMPERATURE	DONOR_BLOODPRESSURE
1114882572	Jetana	Abudawood	57	36	110
1117864572	Elan	Abudawood	55	36.5	120
1109828666	Hadeel	Althbiti	55	36.5	150
1000000001	RIMA	ALGHAMDI	60.66	37.33	166
1112346545	Talya	Abudawood	60	37	130
1119959653	Dhari	Althbiti	55	36.5	150
1000000002	ALI	ALGHAMDI	70.77	37.44	177
1116354851	Hasna	Bukhari	62.59	37	115

8.6 Update Example

Update in Natural Language (English)

Update blood type to AB- for the donee with file number = 114

SQL Script

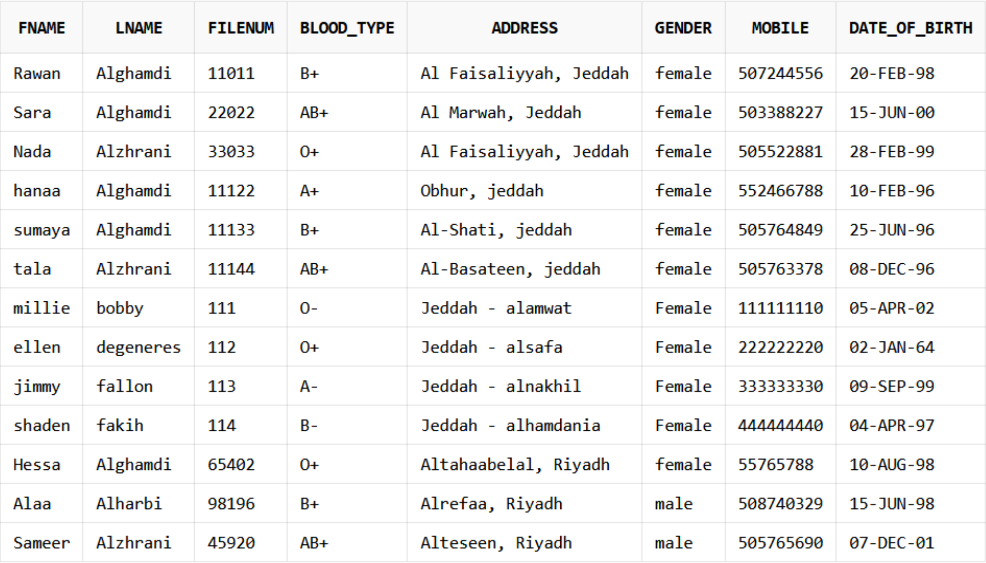
```
UPDATE Donee

SET BLOOD_TYPE = 'AB-'

WHERE FILENUM = 114;
```

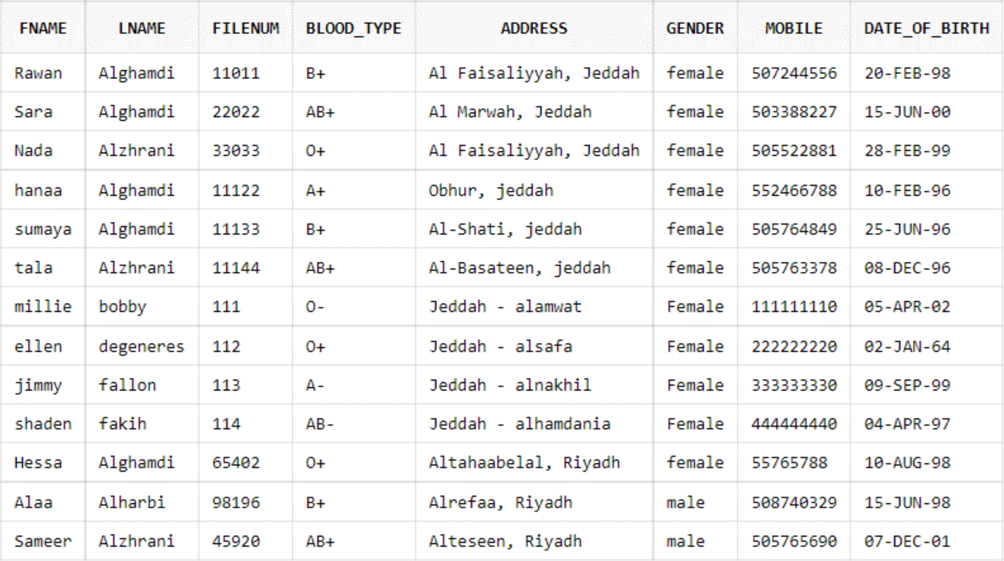
Caption of the Output

Before updating:



FNAME	LNAME	FILENUM	BLOOD_TYPE	ADDRESS	GENDER	MOBILE	DATE_OF_BIRTH
Rawan	Alghamdi	11011	B+	Al Faisaliyyah, Jeddah	female	507244556	20-FEB-98
Sara	Alghamdi	22022	AB+	Al Marwah, Jeddah	female	503388227	15-JUN-00
Nada	Alzhrani	33033	O+	Al Faisaliyyah, Jeddah	female	505522881	28-FEB-99
hanaa	Alghamdi	11122	A+	Obhur, jeddah	female	552466788	10-FEB-96
sumaya	Alghamdi	11133	B+	Al-Shati, jeddah	female	505764849	25-JUN-96
tala	Alzhrani	11144	AB+	Al-Basateen, jeddah	female	505763378	08-DEC-96
millie	bobby	111	O-	Jeddah - alamwat	Female	111111110	05-APR-02
ellen	degeneres	112	O+	Jeddah - alsafa	Female	222222220	02-JAN-64
jimmy	fallon	113	A-	Jeddah - alnakhil	Female	333333330	09-SEP-99
shaden	fakih	114	B-	Jeddah - alhamdania	Female	444444440	04-APR-97
Hessa	Alghamdi	65402	O+	Altahaabelal, Riyadh	female	55765788	10-AUG-98
Alaa	Alharbi	98196	B+	Alrefaa, Riyadh	male	508740329	15-JUN-98
Sameer	Alzhrani	45920	AB+	Alteseen, Riyadh	male	505765690	07-DEC-01

After updating:



FNAME	LNAME	FILENUM	BLOOD_TYPE	ADDRESS	GENDER	MOBILE	DATE_OF_BIRTH
Rawan	Alghamdi	11011	B+	Al Faisaliyyah, Jeddah	female	507244556	20-FEB-98
Sara	Alghamdi	22022	AB+	Al Marwah, Jeddah	female	503388227	15-JUN-00
Nada	Alzhrani	33033	O+	Al Faisaliyyah, Jeddah	female	505522881	28-FEB-99
hanaa	Alghamdi	11122	A+	Obhur, jeddah	female	552466788	10-FEB-96
sumaya	Alghamdi	11133	B+	Al-Shati, jeddah	female	505764849	25-JUN-96
tala	Alzhrani	11144	AB+	Al-Basateen, jeddah	female	505763378	08-DEC-96
millie	bobby	111	O-	Jeddah - alamwat	Female	111111110	05-APR-02
ellen	degeneres	112	O+	Jeddah - alsafa	Female	222222220	02-JAN-64
jimmy	fallon	113	A-	Jeddah - alnakhil	Female	333333330	09-SEP-99
shaden	fakih	114	AB-	Jeddah - alhamdania	Female	444444440	04-APR-97
Hessa	Alghamdi	65402	O+	Altahaabelal, Riyadh	female	55765788	10-AUG-98
Alaa	Alharbi	98196	B+	Alrefaa, Riyadh	male	508740329	15-JUN-98
Sameer	Alzhrani	45920	AB+	Alteseen, Riyadh	male	505765690	07-DEC-01

8.7 Delete Example

Delete in Natural Language (English)

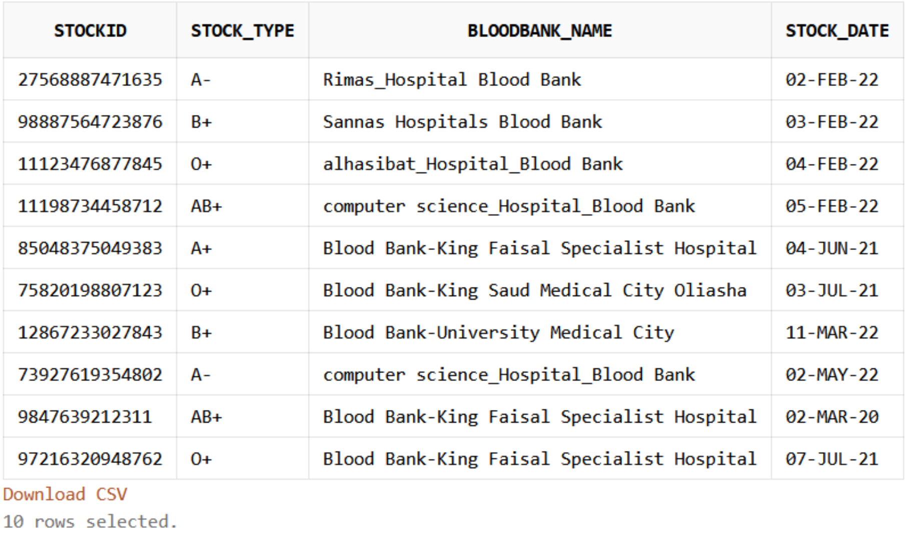
Deleting all B+ stocks.

SQL Script

```
DELETE FROM Blood_Stock
WHERE STOCK_TYPE = 'B+';
```

Caption of the Output

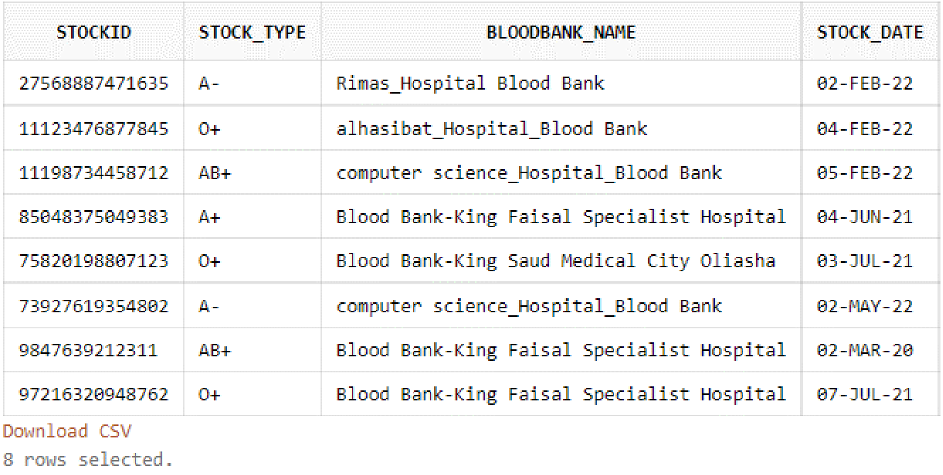
Before deleting:



STOCKID	STOCK_TYPE	BLOODBANK_NAME	STOCK_DATE
27568887471635	A-	Rimas_Hospital Blood Bank	02-FEB-22
98887564723876	B+	Sannas Hospitals Blood Bank	03-FEB-22
11123476877845	O+	alhasibat_Hospital_Blood Bank	04-FEB-22
11198734458712	AB+	computer science_Hospital_Blood Bank	05-FEB-22
85048375049383	A+	Blood Bank-King Faisal Specialist Hospital	04-JUN-21
75820198807123	O+	Blood Bank-King Saud Medical City Oliasha	03-JUL-21
12867233027843	B+	Blood Bank-University Medical City	11-MAR-22
73927619354802	A-	computer science_Hospital_Blood Bank	02-MAY-22
9847639212311	AB+	Blood Bank-King Faisal Specialist Hospital	02-MAR-20
97216320948762	O+	Blood Bank-King Faisal Specialist Hospital	07-JUL-21

Download CSV
10 rows selected.

After deleting:

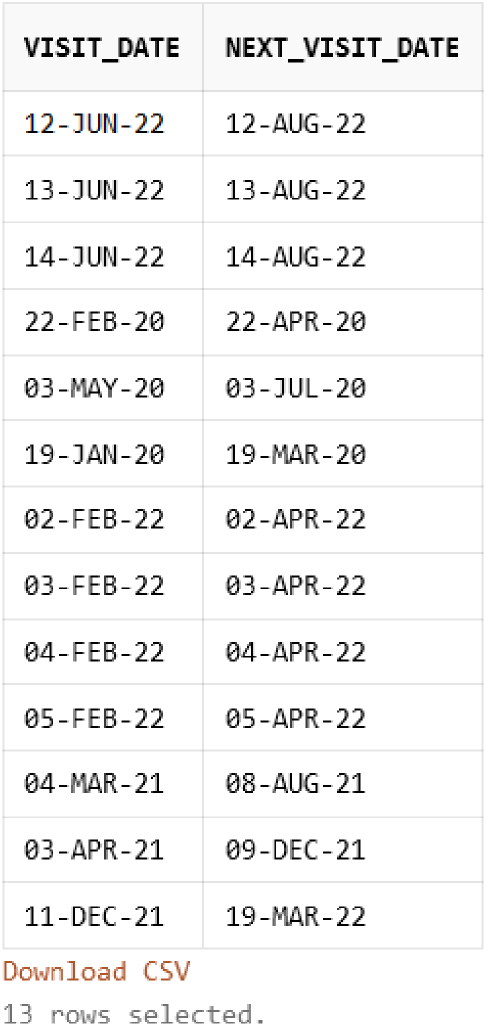


STOCKID	STOCK_TYPE	BLOODBANK_NAME	STOCK_DATE
27568887471635	A-	Rimas_Hospital Blood Bank	02-FEB-22
11123476877845	O+	alhasibat_Hospital_Blood Bank	04-FEB-22
11198734458712	AB+	computer science_Hospital_Blood Bank	05-FEB-22
85048375049383	A+	Blood Bank-King Faisal Specialist Hospital	04-JUN-21
75820198807123	O+	Blood Bank-King Saud Medical City Oliasha	03-JUL-21
73927619354802	A-	computer science_Hospital_Blood Bank	02-MAY-22
9847639212311	AB+	Blood Bank-King Faisal Specialist Hospital	02-MAR-20
97216320948762	O+	Blood Bank-King Faisal Specialist Hospital	07-JUL-21

Download CSV
8 rows selected.

APPENDIX

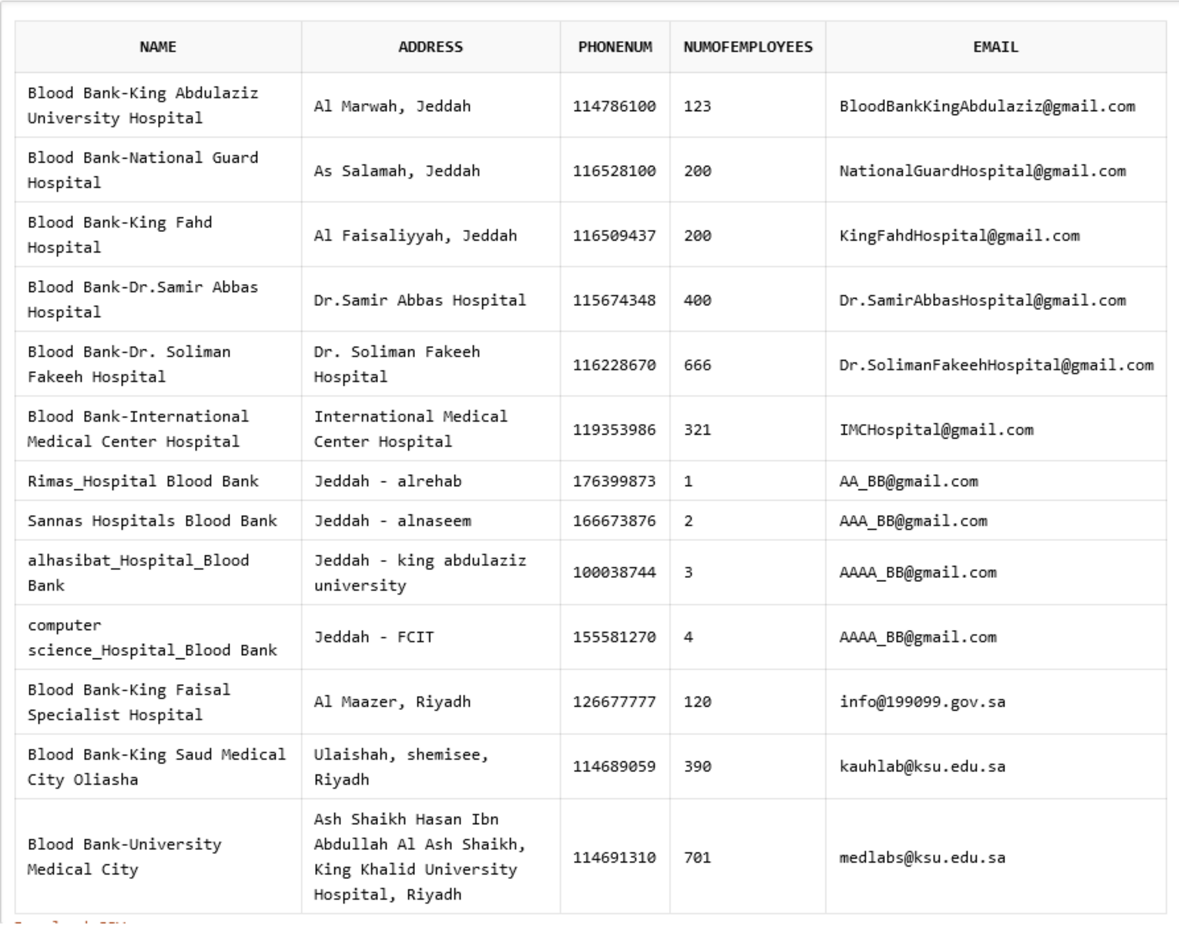
< Visits > TABLE



VISIT_DATE	NEXT_VISIT_DATE
12-JUN-22	12-AUG-22
13-JUN-22	13-AUG-22
14-JUN-22	14-AUG-22
22-FEB-20	22-APR-20
03-MAY-20	03-JUL-20
19-JAN-20	19-MAR-20
02-FEB-22	02-APR-22
03-FEB-22	03-APR-22
04-FEB-22	04-APR-22
05-FEB-22	05-APR-22
04-MAR-21	08-AUG-21
03-APR-21	09-DEC-21
11-DEC-21	19-MAR-22

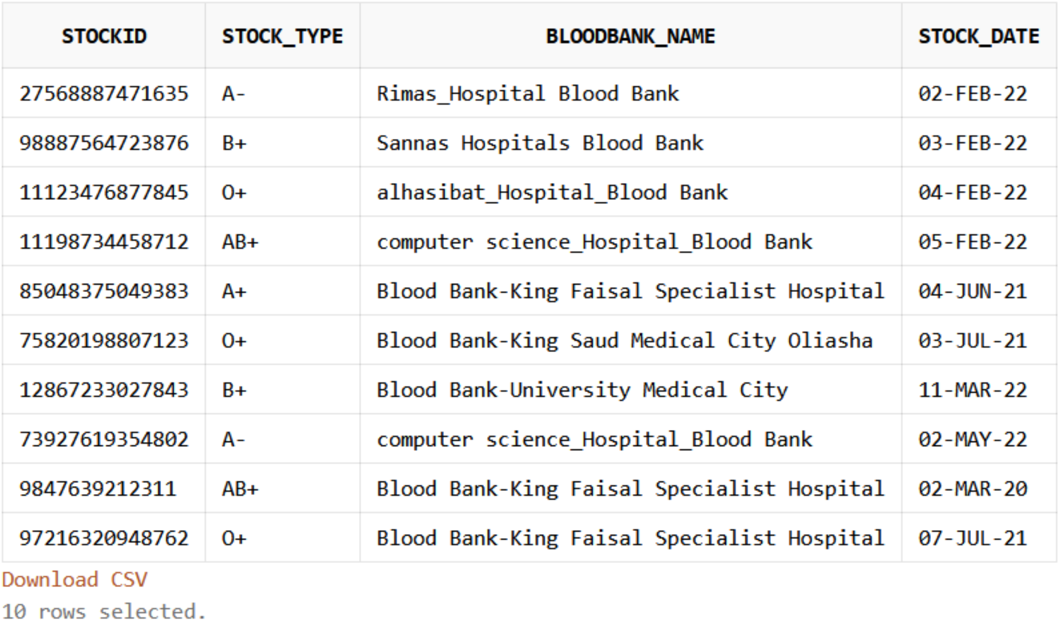
```
Download CSV
13 rows selected.
```

< Blood_Bank > TABLE



NAME	ADDRESS	PHONENUM	NUMOFEMPLOYEES	EMAIL
Blood Bank-King Abdulaziz University Hospital	Al Marwah, Jeddah	114786100	123	BloodBankKingAbdulaziz@gmail.com
Blood Bank-National Guard Hospital	As Salamah, Jeddah	116528100	200	NationalGuardHospital@gmail.com
Blood Bank-King Fahd Hospital	Al Faisaliyyah, Jeddah	116509437	200	KingFahdHospital@gmail.com
Blood Bank-Dr.Samir Abbas Hospital	Dr.Samir Abbas Hospital	115674348	400	Dr.SamirAbbasHospital@gmail.com
Blood Bank-Dr. Soliman Fakeeh Hospital	Dr. Soliman Fakeeh Hospital	116228670	666	Dr.SolimanFakeehHospital@gmail.com
Blood Bank-International Medical Center Hospital	International Medical Center Hospital	119353986	321	IMCHospital@gmail.com
Rimas_Hospital Blood Bank	Jeddah - alrehab	176399873	1	AA_BB@gmail.com
Sannas Hospitals Blood Bank	Jeddah - alnaseem	166673876	2	AAA_BB@gmail.com
alhasibat_Hospital_Blood Bank	Jeddah - king abdulaziz university	100038744	3	AAAA_BB@gmail.com
computer science_Hospital_Blood Bank	Jeddah - FCIT	155581270	4	AAAA_BB@gmail.com
Blood Bank-King Faisal Specialist Hospital	Al Maazer, Riyadh	126677777	120	info@199099.gov.sa
Blood Bank-King Saud Medical City Oliasha	Ulaishah, shemisee, Riyadh	114689059	390	kauhlab@ksu.edu.sa
Blood Bank-University Medical City	Ash Shaikh Hasan Ibn Abdullah Al Ash Shaikh, King Khalid University Hospital, Riyadh	114691310	701	medlabs@ksu.edu.sa

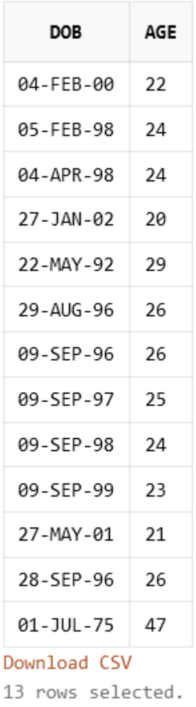
< Blood_Stock > TABLE



STOCKID	STOCK_TYPE	BLOODBANK_NAME	STOCK_DATE
27568887471635	A-	Rimas_Hospital Blood Bank	02-FEB-22
98887564723876	B+	Sannas Hospitals Blood Bank	03-FEB-22
11123476877845	O+	alhasibat_Hospital_Blood Bank	04-FEB-22
11198734458712	AB+	computer science_Hospital_Blood Bank	05-FEB-22
85048375049383	A+	Blood Bank-King Faisal Specialist Hospital	04-JUN-21
75820198807123	O+	Blood Bank-King Saud Medical City Oliasha	03-JUL-21
12867233027843	B+	Blood Bank-University Medical City	11-MAR-22
73927619354802	A-	computer science_Hospital_Blood Bank	02-MAY-22
9847639212311	AB+	Blood Bank-King Faisal Specialist Hospital	02-MAR-20
97216320948762	O+	Blood Bank-King Faisal Specialist Hospital	07-JUL-21

Download CSV
10 rows selected.

< Donor_age > TABLE

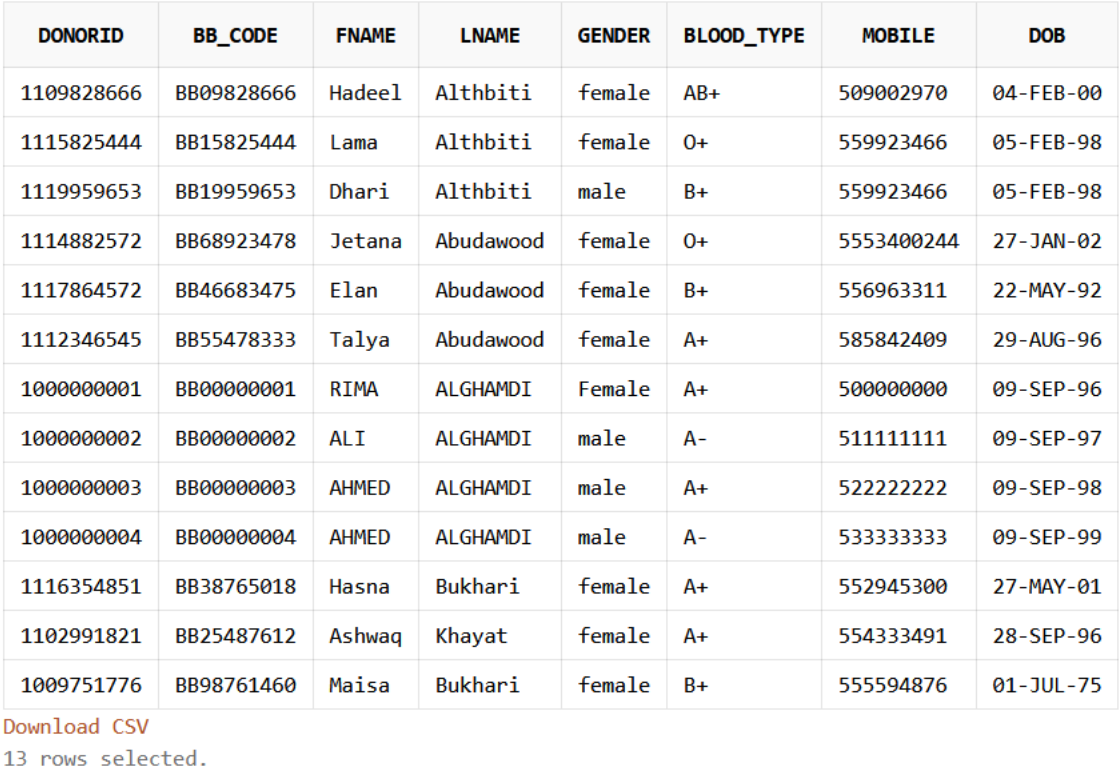


DOB	AGE
04-FEB-00	22
05-FEB-98	24
04-APR-98	24
27-JAN-02	20
22-MAY-92	29
29-AUG-96	26
09-SEP-96	26
09-SEP-97	25
09-SEP-98	24
09-SEP-99	23
27-MAY-01	21
28-SEP-96	26
01-JUL-75	47

Download CSV

13 rows selected.

< Donor > TABLE

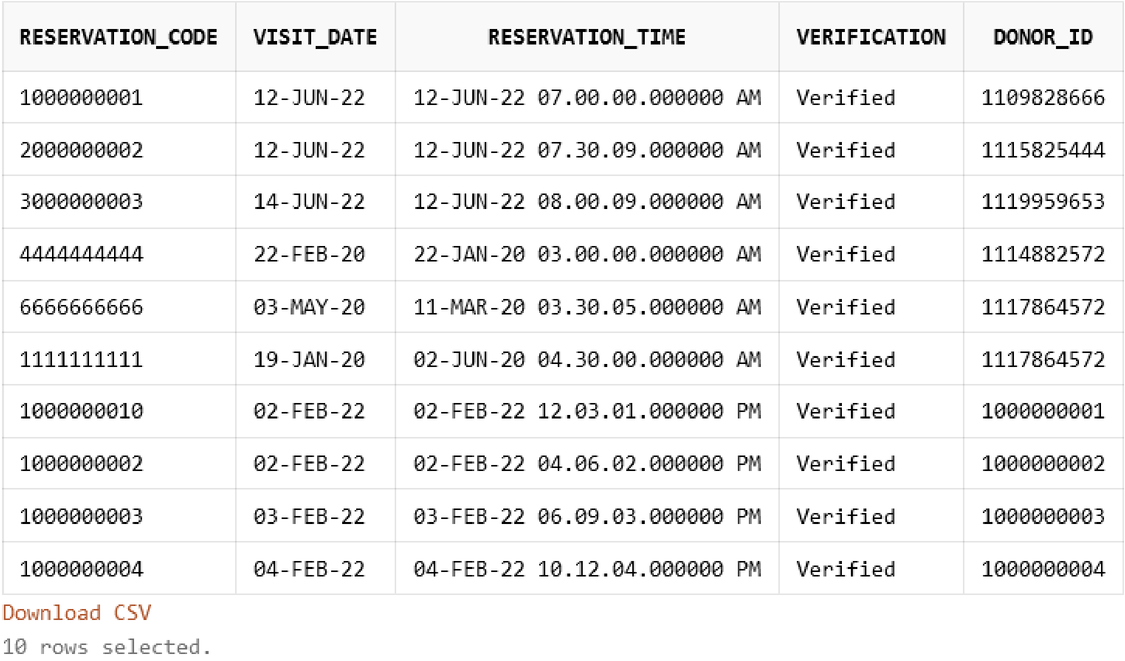


DONORID	BB_CODE	FNAME	LNAME	GENDER	BLOOD_TYPE	MOBILE	DOB
1109828666	BB09828666	Hadeel	Althbiti	female	AB+	509002970	04-FEB-00
1115825444	BB15825444	Lama	Althbiti	female	O+	559923466	05-FEB-98
1119959653	BB19959653	Dhari	Althbiti	male	B+	559923466	05-FEB-98
1114882572	BB68923478	Jetana	Abudawood	female	O+	5553400244	27-JAN-02
1117864572	BB46683475	Elan	Abudawood	female	B+	556963311	22-MAY-92
1112346545	BB55478333	Talya	Abudawood	female	A+	585842409	29-AUG-96
1000000001	BB00000001	RIMA	ALGHAMDI	Female	A+	500000000	09-SEP-96
1000000002	BB00000002	ALI	ALGHAMDI	male	A-	511111111	09-SEP-97
1000000003	BB00000003	AHMED	ALGHAMDI	male	A+	522222222	09-SEP-98
1000000004	BB00000004	AHMED	ALGHAMDI	male	A-	533333333	09-SEP-99
1116354851	BB38765018	Hasna	Bukhari	female	A+	552945300	27-MAY-01
1102991821	BB25487612	Ashwaq	Khayat	female	A+	554333491	28-SEP-96
1009751776	BB98761460	Maisa	Bukhari	female	B+	555594876	01-JUL-75

Download CSV

13 rows selected.

< Reservation > TABLE

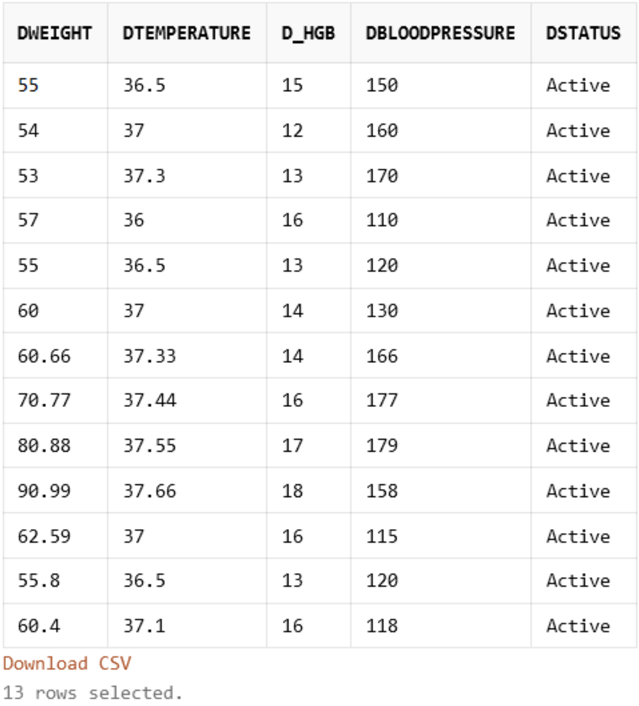


RESERVATION_CODE	VISIT_DATE	RESERVATION_TIME	VERIFICATION	DONOR_ID
1000000001	12-JUN-22	12-JUN-22 07.00.00.000000 AM	Verified	1109828666
2000000002	12-JUN-22	12-JUN-22 07.30.09.000000 AM	Verified	1115825444
3000000003	14-JUN-22	12-JUN-22 08.00.09.000000 AM	Verified	1119959653
4444444444	22-FEB-20	22-JAN-20 03.00.00.000000 AM	Verified	1114882572
6666666666	03-MAY-20	11-MAR-20 03.30.05.000000 AM	Verified	1117864572
1111111111	19-JAN-20	02-JUN-20 04.30.00.000000 AM	Verified	1117864572
1000000010	02-FEB-22	02-FEB-22 12.03.01.000000 PM	Verified	1000000001
1000000002	02-FEB-22	02-FEB-22 04.06.02.000000 PM	Verified	1000000002
1000000003	03-FEB-22	03-FEB-22 06.09.03.000000 PM	Verified	1000000003
1000000004	04-FEB-22	04-FEB-22 10.12.04.000000 PM	Verified	1000000004

Download CSV

10 rows selected.

< Donor_Status > TABLE

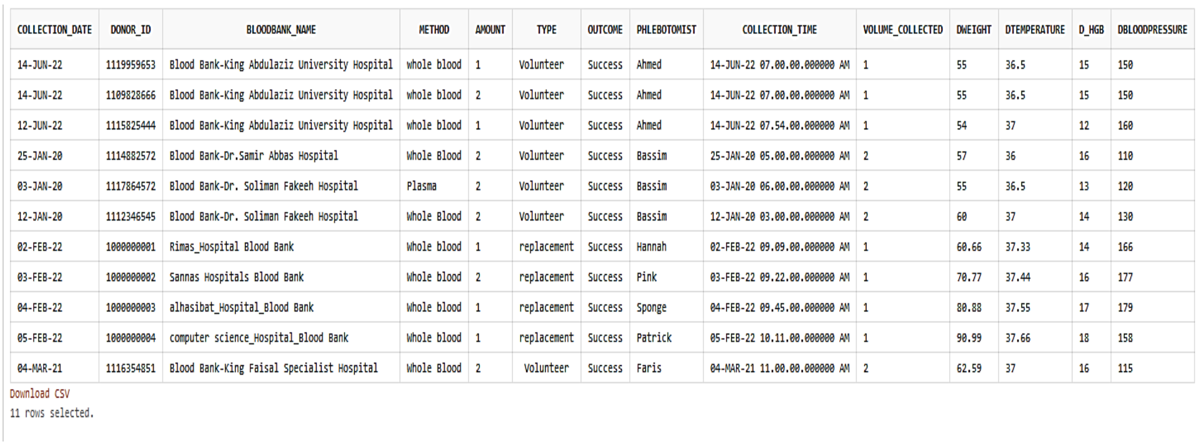


DWEIGHT	DTEMPERATURE	D_HGB	DBLOODPRESSURE	DSTATUS
55	36.5	15	150	Active
54	37	12	160	Active
53	37.3	13	170	Active
57	36	16	110	Active
55	36.5	13	120	Active
60	37	14	130	Active
60.66	37.33	14	166	Active
70.77	37.44	16	177	Active
80.88	37.55	17	179	Active
90.99	37.66	18	158	Active
62.59	37	16	115	Active
55.8	36.5	13	120	Active
60.4	37.1	16	118	Active

Download CSV

13 rows selected.

< Donation > TABLE

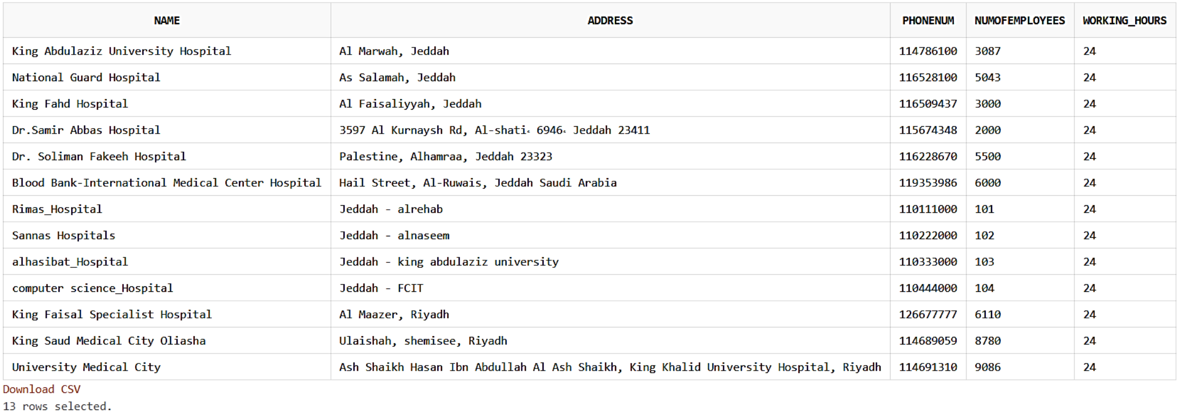


COLLECTION_DATE	DONOR_ID	BLOODBANK_NAME	METHOD	AMOUNT	TYPE	OUTCOME	PHLEBOTOMIST	COLLECTION_TIME	VOLUME_COLLECTED	DWEIGHT	DTEMPERATURE	D_HGB	DBLOODPRESSURE
14-JUN-22	1119959653	Blood Bank-King Abdulaziz University Hospital	whole blood	1	Volunteer	Success	Ahmed	14-JUN-22 07.00.00.000000 AM	1	55	36.5	15	150
14-JUN-22	1109828666	Blood Bank-King Abdulaziz University Hospital	whole blood	2	Volunteer	Success	Ahmed	14-JUN-22 07.00.00.000000 AM	1	55	36.5	15	150
12-JUN-22	1115825444	Blood Bank-King Abdulaziz University Hospital	whole blood	1	Volunteer	Success	Ahmed	14-JUN-22 07.54.00.000000 AM	1	54	37	12	160
25-JAN-20	1114882572	Blood Bank-Dr.Samir Abbas Hospital	Whole Blood	2	Volunteer	Success	Bassim	25-JAN-20 05.00.00.000000 AM	2	57	36	16	110
03-JAN-20	1117864572	Blood Bank-Dr. Soliman Fakeeh Hospital	Plasma	2	Volunteer	Success	Bassim	03-JAN-20 06.00.00.000000 AM	2	55	36.5	13	120
12-JAN-20	1112346545	Blood Bank-Dr. Soliman Fakeeh Hospital	Whole Blood	2	Volunteer	Success	Bassim	12-JAN-20 03.00.00.000000 AM	2	60	37	14	130
02-FEB-22	1000000001	Rimas_Hospital Blood Bank	Whole blood	1	replacement	Success	Hannah	02-FEB-22 09.09.00.000000 AM	1	60.66	37.33	14	166
03-FEB-22	1000000002	Sannas Hospitals Blood Bank	Whole blood	2	replacement	Success	Pink	03-FEB-22 09.22.00.000000 AM	1	70.77	37.44	16	177
04-FEB-22	1000000003	alhasibat_Hospital_Blood Bank	Whole blood	1	replacement	Success	Sponge	04-FEB-22 09.45.00.000000 AM	1	80.88	37.55	17	179
05-FEB-22	1000000004	computer science_Hospital_Blood Bank	Whole blood	1	replacement	Success	Patrick	05-FEB-22 10.11.00.000000 AM	1	90.99	37.66	18	158
04-MAR-21	1116354851	Blood Bank-King Faisal Specialist Hospital	Whole Blood	2	Volunteer	Success	Faris	04-MAR-21 11.00.00.000000 AM	2	62.59	37	16	115

Download CSV

11 rows selected.

< Hospital > TABLE

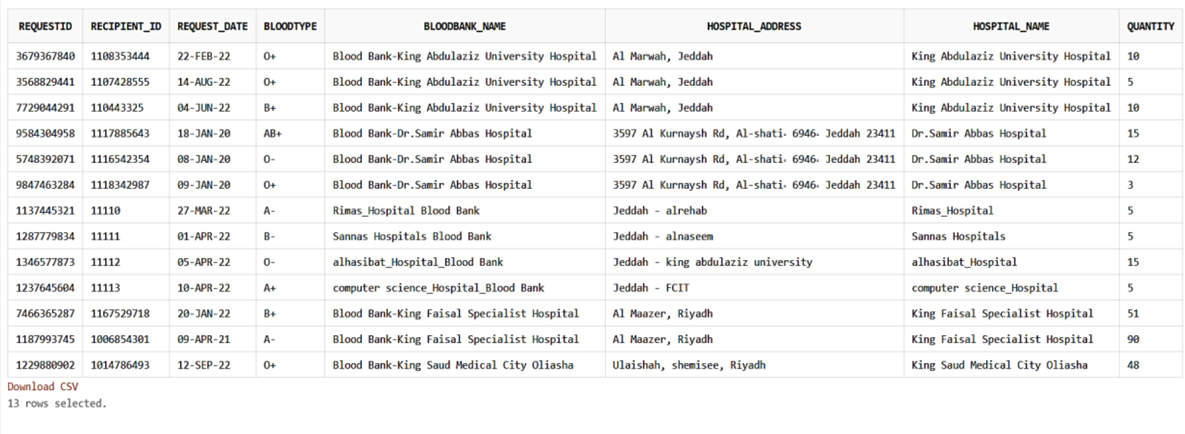


NAME	ADDRESS	PHONENUM	NUMOFEMPLOYEES	WORKING_HOURS
King Abdulaziz University Hospital	Al Marwah, Jeddah	114786100	3087	24
National Guard Hospital	As Salamah, Jeddah	116528100	5043	24
King Fahd Hospital	Al Faisaliyyah, Jeddah	116509437	3000	24
Dr.Samir Abbas Hospital	3597 Al Kurnaysh Rd, Al-shati، 6946، Jeddah 23411	115674348	2000	24
Dr. Soliman Fakeeh Hospital	Palestine, Alhamraa, Jeddah 23323	116228670	5500	24
Blood Bank-International Medical Center Hospital	Hail Street, Al-Ruwais, Jeddah Saudi Arabia	119353986	6000	24
Rimas_Hospital	Jeddah - alrehab	110111000	101	24
Sannas Hospitals	Jeddah - alnaseem	110222000	102	24
alhasibat_Hospital	Jeddah - king abdulaziz university	110333000	103	24
computer science_Hospital	Jeddah - FCIT	110444000	104	24
King Faisal Specialist Hospital	Al Maazer, Riyadh	126677777	6110	24
King Saud Medical City Oliasha	Ulaishah, shemisee, Riyadh	114689059	8780	24
University Medical City	Ash Shaikh Hasan Ibn Abdullah Al Ash Shaikh, King Khalid University Hospital, Riyadh	114691310	9086	24

Download CSV

13 rows selected.

< Request > TABLE

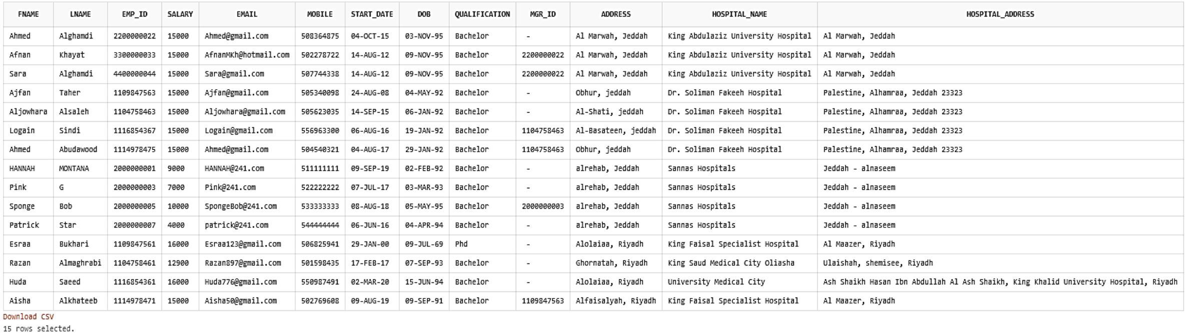


REQUESTID	RECIPIENT_ID	REQUEST_DATE	BLOODTYPE	BLOODBANK_NAME	HOSPITAL_ADDRESS	HOSPITAL_NAME	QUANTITY
3679367840	1108353444	22-FEB-22	O+	Blood Bank-King Abdulaziz University Hospital	Al Marwah, Jeddah	King Abdulaziz University Hospital	10
3568829441	1107428555	14-AUG-22	O+	Blood Bank-King Abdulaziz University Hospital	Al Marwah, Jeddah	King Abdulaziz University Hospital	5
7729044291	110443325	04-JUN-22	B+	Blood Bank-King Abdulaziz University Hospital	Al Marwah, Jeddah	King Abdulaziz University Hospital	10
9584304958	1117885643	18-JAN-20	AB+	Blood Bank-Dr.Samir Abbas Hospital	3597 Al Kurnaysh Rd, Al-shati، 6946، Jeddah 23411	Dr.Samir Abbas Hospital	15
5748392071	1116542354	08-JAN-20	O-	Blood Bank-Dr.Samir Abbas Hospital	3597 Al Kurnaysh Rd, Al-shati، 6946، Jeddah 23411	Dr.Samir Abbas Hospital	12
9847463284	1118342987	09-JAN-20	O+	Blood Bank-Dr.Samir Abbas Hospital	3597 Al Kurnaysh Rd, Al-shati، 6946، Jeddah 23411	Dr.Samir Abbas Hospital	3
1137445321	11110	27-MAR-22	A-	Rimas_Hospital Blood Bank	Jeddah - alrehab	Rimas_Hospital	5
1287779834	11111	01-APR-22	B-	Sannas Hospitals Blood Bank	Jeddah - alnaseem	Sannas Hospitals	5
1346577873	11112	05-APR-22	O-	alhasibat_Hospital_Blood Bank	Jeddah - king abdulaziz university	alhasibat_Hospital	15
1237645604	11113	10-APR-22	A+	computer science_Hospital_Blood Bank	Jeddah - FCIT	computer science_Hospital	5
7466365287	1167529718	20-JAN-22	B+	Blood Bank-King Faisal Specialist Hospital	Al Maazer, Riyadh	King Faisal Specialist Hospital	51
1187993745	1006854301	09-APR-21	A-	Blood Bank-King Faisal Specialist Hospital	Al Maazer, Riyadh	King Faisal Specialist Hospital	90
1229880902	1014786493	12-SEP-22	O+	Blood Bank-King Saud Medical City Oliasha	Ulaishah, shemisee, Riyadh	King Saud Medical City Oliasha	48

Download CSV

13 rows selected.

< Employee > TABLE

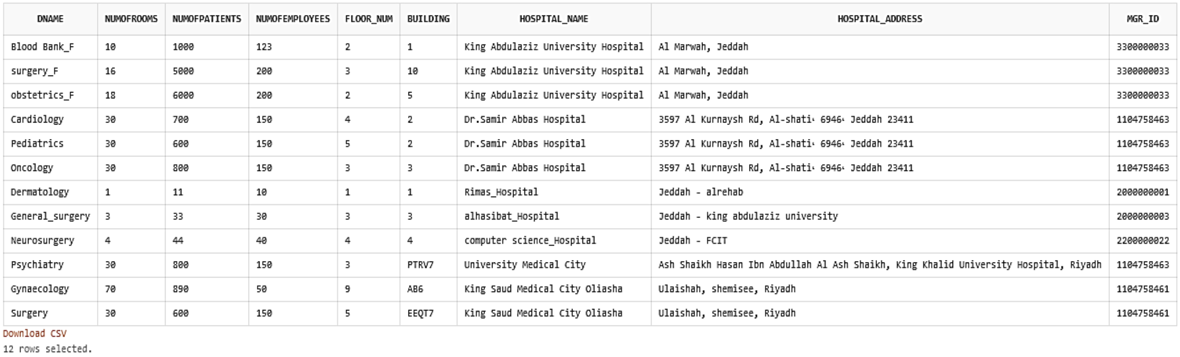


FNAME	LNAME	EMP_ID	SALARY	EMAIL	MOBILE	START_DATE	DOB	QUALIFICATION	MGR_ID	ADDRESS	HOSPITAL_NAME	HOSPITAL_ADDRESS
Ahmed	Alghamdi	2200000022	15000	Ahmed@gmail.com	508364875	04-OCT-15	03-NOV-95	Bachelor	-	Al Marwah, Jeddah	King Abdulaziz University Hospital	Al Marwah, Jeddah
Afnan	Khayat	3300000033	15000	AfnanMKh@hotmail.com	502278722	14-AUG-12	09-NOV-95	Bachelor	2200000022	Al Marwah, Jeddah	King Abdulaziz University Hospital	Al Marwah, Jeddah
Sara	Alghamdi	4400000044	15000	Sara@gmail.com	507744338	14-AUG-12	09-NOV-95	Bachelor	2200000022	Al Marwah, Jeddah	King Abdulaziz University Hospital	Al Marwah, Jeddah
Ajfan	Taher	1109847563	15000	Ajfan@gmail.com	505340098	24-AUG-08	04-MAY-92	Bachelor	-	Obhur, jeddah	Dr. Soliman Fakeeh Hospital	Palestine, Alhamraa, Jeddah 23323
Aljowhara	Alsaleh	1104758463	15000	Aljowhara@gmail.com	505623035	14-SEP-15	06-JAN-92	Bachelor	-	Al-Shati, jeddah	Dr. Soliman Fakeeh Hospital	Palestine, Alhamraa, Jeddah 23323
Logain	Sindi	1116854367	15000	Logain@gmail.com	556963300	06-AUG-16	19-JAN-92	Bachelor	1104758463	Al-Basateen, jeddah	Dr. Soliman Fakeeh Hospital	Palestine, Alhamraa, Jeddah 23323
Ahmed	Abudawood	1114978475	15000	Ahmed@gmail.com	504540321	04-AUG-17	29-JAN-92	Bachelor	1104758463	Obhur, jeddah	Dr. Soliman Fakeeh Hospital	Palestine, Alhamraa, Jeddah 23323
HANNAH	MONTANA	2000000001	9000	HANNAH@241.com	511111111	09-SEP-19	02-FEB-92	Bachelor	-	alrehab, Jeddah	Sannas Hospitals	Jeddah - alnaseem
Pink	G	2000000003	7000	Pink@241.com	522222222	07-JUL-17	03-MAR-93	Bachelor	-	alrehab, Jeddah	Sannas Hospitals	Jeddah - alnaseem
Sponge	Bob	2000000005	10000	SpongeBob@241.com	533333333	08-AUG-18	05-MAY-95	Bachelor	2000000003	alrehab, Jeddah	Sannas Hospitals	Jeddah - alnaseem
Patrick	Star	2000000007	4000	patrick@241.com	544444444	06-JUN-16	04-APR-94	Bachelor	-	alrehab, Jeddah	Sannas Hospitals	Jeddah - alnaseem
Esraa	Bukhari	1109847561	16000	Esraa123@gmail.com	506825941	29-JAN-00	09-JUL-69	Phd	-	Alolaiaa, Riyadh	King Faisal Specialist Hospital	Al Maazer, Riyadh
Razan	Almaghrabi	1104758461	12900	Razan897@gmail.com	501598435	17-FEB-17	07-SEP-93	Bachelor	-	Ghornatah, Riyadh	King Saud Medical City Oliasha	Ulaishah, shemisee, Riyadh
Huda	Saeed	1116854361	16000	Huda776@gmail.com	550987491	02-MAR-20	15-JUN-94	Bachelor	-	Alolaiaa, Riyadh	University Medical City	Ash Shaikh Hasan Ibn Abdullah Al Ash Shaikh, King Khalid University Hospital, Riyadh
Aisha	Alkhateeb	1114978471	15000	Aisha50@gmail.com	502769608	09-AUG-19	09-SEP-91	Bachelor	1109847563	Alfaisalyah, Riyadh	King Faisal Specialist Hospital	Al Maazer, Riyadh

Download CSV

15 rows selected.

< Department > TABLE

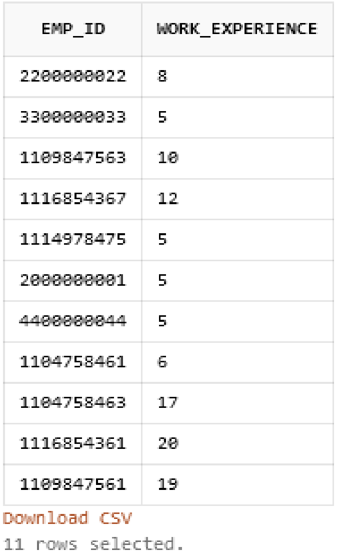


DNAME	NUMOFROOMS	NUMOFPATIENTS	NUMOFEMPLOYEES	FLOOR_NUM	BUILDING	HOSPITAL_NAME	HOSPITAL_ADDRESS	MGR_ID
Blood Bank_F	10	1000	123	2	1	King Abdulaziz University Hospital	Al Marwah, Jeddah	3300000033
surgery_F	16	5000	200	3	10	King Abdulaziz University Hospital	Al Marwah, Jeddah	3300000033
obstetrics_F	18	6000	200	2	5	King Abdulaziz University Hospital	Al Marwah, Jeddah	3300000033
Cardiology	30	700	150	4	2	Dr.Samir Abbas Hospital	3597 Al Kurnaysh Rd, Al-shati، 6946، Jeddah 23411	1104758463
Pediatrics	30	600	150	5	2	Dr.Samir Abbas Hospital	3597 Al Kurnaysh Rd, Al-shati، 6946، Jeddah 23411	1104758463
Oncology	30	800	150	3	3	Dr.Samir Abbas Hospital	3597 Al Kurnaysh Rd, Al-shati، 6946، Jeddah 23411	1104758463
Dermatology	1	11	10	1	1	Rimas_Hospital	Jeddah - alrehab	2000000001
General_surgery	3	33	30	3	3	alhasibat_Hospital	Jeddah - king abdulaziz university	2000000003
Neurosurgery	4	44	40	4	4	computer science_Hospital	Jeddah - FCIT	2200000022
Psychiatry	30	800	150	3	PTRV7	University Medical City	Ash Shaikh Hasan Ibn Abdullah Al Ash Shaikh, King Khalid University Hospital, Riyadh	1104758463
Gynaecology	70	890	50	9	AB6	King Saud Medical City Oliasha	Ulaishah, shemisee, Riyadh	1104758461
Surgery	30	600	150	5	EEQT7	King Saud Medical City Oliasha	Ulaishah, shemisee, Riyadh	1104758461

Download CSV

12 rows selected.

< Manager > TABLE

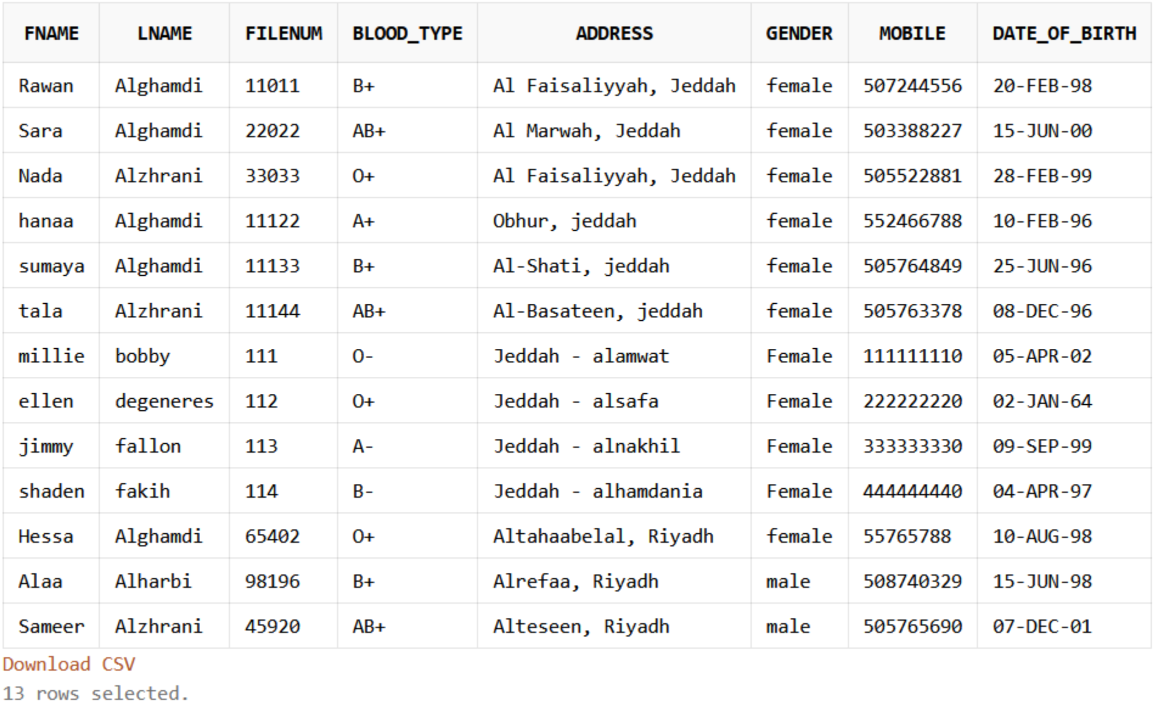


EMP_ID	WORK_EXPERIENCE
2200000022	8
3300000033	5
1109847563	10
1116854367	12
1114978475	5
2000000001	5
4400000044	5
1104758461	6
1104758463	17
1116854361	20
1109847561	19

Download CSV

11 rows selected.

< Donee > TABLE

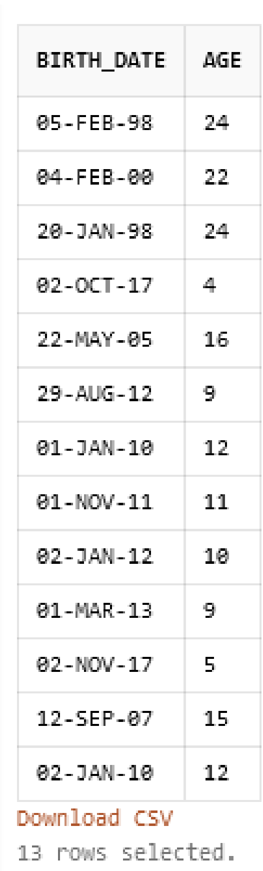


FNAME	LNAME	FILENUM	BLOOD_TYPE	ADDRESS	GENDER	MOBILE	DATE_OF_BIRTH
Rawan	Alghamdi	11011	B+	Al Faisaliyyah, Jeddah	female	507244556	20-FEB-98
Sara	Alghamdi	22022	AB+	Al Marwah, Jeddah	female	503388227	15-JUN-00
Nada	Alzhrani	33033	O+	Al Faisaliyyah, Jeddah	female	505522881	28-FEB-99
hanaa	Alghamdi	11122	A+	Obhur, jeddah	female	552466788	10-FEB-96
sumaya	Alghamdi	11133	B+	Al-Shati, jeddah	female	505764849	25-JUN-96
tala	Alzhrani	11144	AB+	Al-Basateen, jeddah	female	505763378	08-DEC-96
millie	bobby	111	O-	Jeddah - alamwat	Female	111111110	05-APR-02
ellen	degeneres	112	O+	Jeddah - alsafa	Female	222222220	02-JAN-64
jimmy	fallon	113	A-	Jeddah - alnakhil	Female	333333330	09-SEP-99
shaden	fakih	114	B-	Jeddah - alhamdania	Female	444444440	04-APR-97
Hessa	Alghamdi	65402	O+	Altahaabelal, Riyadh	female	55765788	10-AUG-98
Alaa	Alharbi	98196	B+	Alrefaa, Riyadh	male	508740329	15-JUN-98
Sameer	Alzhrani	45920	AB+	Alteseen, Riyadh	male	505765690	07-DEC-01

Download CSV

13 rows selected.

< Dependent_Ages > TABLE

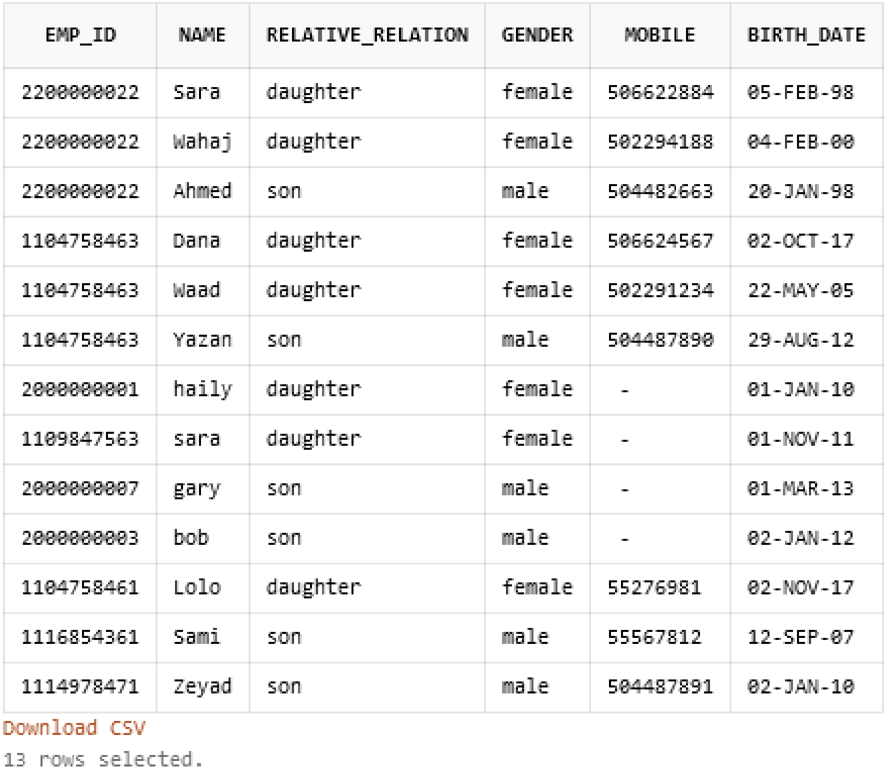


BIRTH_DATE	AGE
05-FEB-98	24
04-FEB-00	22
20-JAN-98	24
02-OCT-17	4
22-MAY-05	16
29-AUG-12	9
01-JAN-10	12
01-NOV-11	11
02-JAN-12	10
01-MAR-13	9
02-NOV-17	5
12-SEP-07	15
02-JAN-10	12

Download CSV

13 rows selected.

< Dependent > TABLE

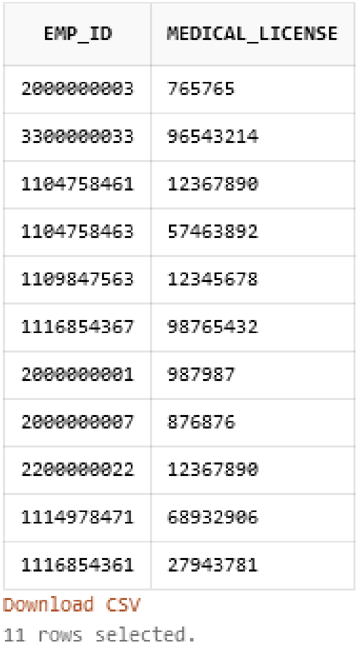


EMP_ID	NAME	RELATIVE_RELATION	GENDER	MOBILE	BIRTH_DATE
2200000022	Sara	daughter	female	506622884	05-FEB-98
2200000022	Wahaj	daughter	female	502294188	04-FEB-00
2200000022	Ahmed	son	male	504482663	20-JAN-98
1104758463	Dana	daughter	female	506624567	02-OCT-17
1104758463	Waad	daughter	female	502291234	22-MAY-05
1104758463	Yazan	son	male	504487890	29-AUG-12
2000000001	haily	daughter	female	-	01-JAN-10
1109847563	sara	daughter	female	-	01-NOV-11
2000000007	gary	son	male	-	01-MAR-13
2000000003	bob	son	male	-	02-JAN-12
1104758461	Lolo	daughter	female	55276981	02-NOV-17
1116854361	Sami	son	male	55567812	12-SEP-07
1114978471	Zeyad	son	male	504487891	02-JAN-10

Download CSV

13 rows selected.

< Health_Practitioner > TABLE

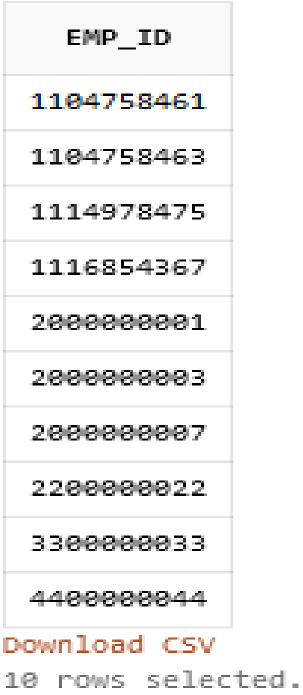


EMP_ID	MEDICAL_LICENSE
2000000003	765765
3300000033	96543214
1104758461	12367890
1104758463	57463892
1109847563	12345678
1116854367	98765432
2000000001	987987
2000000007	876876
2200000022	12367890
1114978471	68932906
1116854361	27943781

Download CSV

11 rows selected.

< Receptionist > TABLE

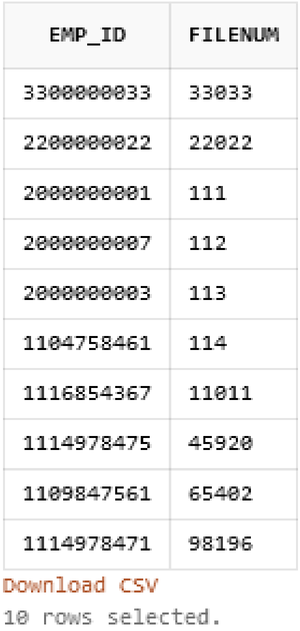


EMP_ID
1104758461
1104758463
1114978475
1116854367
2000000001
2000000003
2000000007
2200000022
3300000033
4400000044

Download CSV

10 rows selected.

< Responsible_For > TABLE

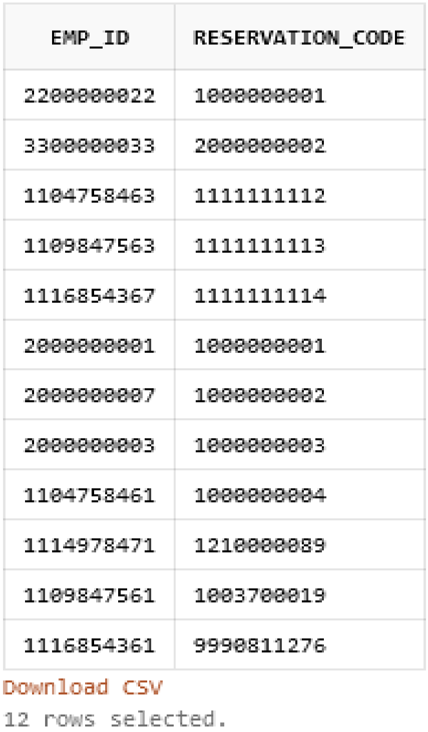


EMP_ID	FILENUM
3300000033	33033
2200000022	22022
2000000001	111
2000000007	112
2000000003	113
1104758461	114
1116854367	11011
1114978475	45920
1109847561	65402
1114978471	98196

Download CSV

10 rows selected.

< Organizes > TABLE



EMP_ID	RESERVATION_CODE
2200000022	1000000001
3300000033	2000000002
1104758463	1111111112
1109847563	1111111113
1116854367	1111111114
2000000001	1000000001
2000000007	1000000002
2000000003	1000000003
1104758461	1000000004
1114978471	1210000089
1109847561	1003700019
1116854361	9990811276

Download CSV

12 rows selected.