राष्ट्रीय प्रौद्योगिकी संस्थान वारंगल
National Institute of Technology Warangal
Department of Computer Science and Engineering

Medical Exigency Database Management System

"A meticulous Database Model of a Medical Exigency Platform implemented in the Structured Query Language (SQL)"

PROBLEM STATEMENT

Project Details

Title	CureGenie (A Medical Emergency-based DataBase)
Creators	Greata Chakravarty(21CSB0B20) Sai Kiran Sajja(21CSB0A49)
Project Type	DBMS Project-1 (SEMESTER I
Under the guidance of	Prof - Dr. R. B. V. Subramaanyam Prof - Dr. T. Ramakrishnudu

Problem Statement

In moments of dire medical exigency, where life hangs in the balance, swift diagnosis and nimble treatment are of paramount import to ward off potential fatal outcomes. A hiatus in either can result in catastrophic consequences, as evidenced by the countless fatalities brought about by the scarcity of life-sustaining injections like Remdesivir and Fabiflu, and the dawdling pace of essential diagnostic tests such as RT-PCR and Rapid Antigen during the Covid-19 crisis. The frantic search for a suitable diagnostic centre by loved ones only adds to the wastage of precious moments that could have been utilized to ensure the patient's survival.

Solution-

CureGenie is a distinctive platform, a veritable treasure trove of medical emergency information, providing a comprehensive database of locations where lifesaving drugs, injections, and medicines can be obtained with utmost speed. It bridges the gap between the patient and the crucial medical services they require, such as tests, diagnostic centers, and injections, by alerting them to the closest available options and thereby preserving precious moments that could mean the difference between life and death. With its nationwide database, the platform displays the locations of the required services, eliminating the need for aimless wanderings or the fear of rejection and reducing stress and anxiety for both patients and their loved ones.

CONTENTS

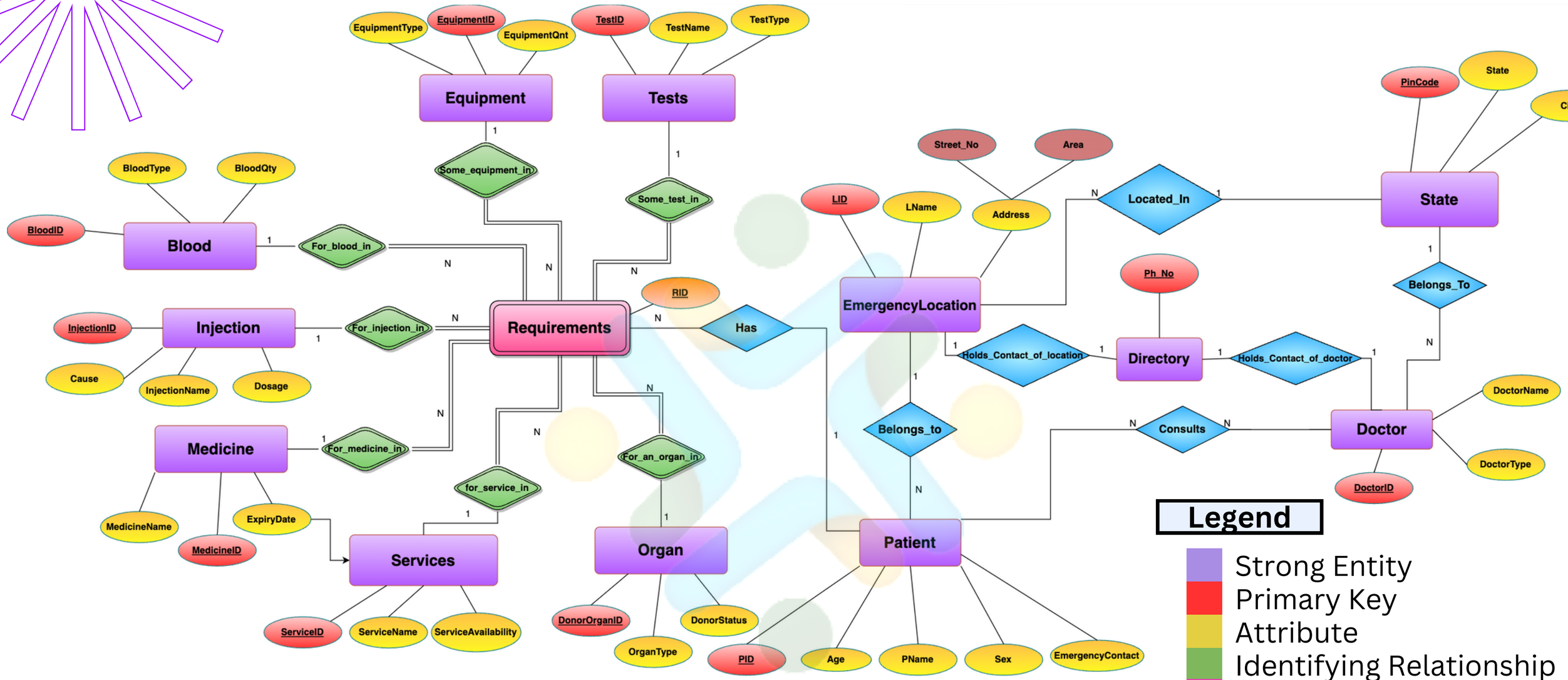
ACKNOWLEDGEMENT

- The attainment of triumph in any enterprise hinges on the potency of inspiration and motivation. A prosperous culmination of a project necessitates efficacious guidance and assistance. We would like to extend our sincerest gratitude to the erudite DBMS professors, **Dr. T Ramakrishnudu** and **Dr. RBV Subramanyam**, for endowing us with the opportunity to delve into the intricate database concepts. Their tutelage has proven to be an invaluable aid in comprehending the fundamental facets of DBMS, encompassing relations, E-R diagrams, Structured Query Language, functional dependencies, normalization, indexing, and more.
- We would also like to convey our heartfelt appreciation to the entire Computer Science and Engineering Department for their amiable and unfailing support whenever we required any form of guidance. The DBMS Lab faculty also deserves a special mention fo nurturing and honing our Structured Query Language skills through their well-crafted and periodically-assigned exercises on Google Classroom.
- The students of CSE batch (II-Year) have contributed immensely to our learning experience with their interactive participation and stimulating class discussions, which have been nothing short of a godsend for all the scholars.
- Furthermore, we express our gratitude to the management of the esteemed National Institute of Technology Warangal, the Dean - Academic, and the Director, for bestowing upon us the privilege to explore this fascinating subject and for opening new avenues in the realm of database management.
- Lastly, we would be remiss if we failed to acknowledge the support and encouragement of our parents, elders, and well-wishers, who have been our unwavering pillars of technical and moral sustenance throughout this academic journey.

Greata Chakravarty
21CSB0B20
(Section-B)

Sai Kiran Sajja
21CSB0A49
(Section-A)

E-R DIAGRAM



Legend

- Strong Entity
- Primary Key
- Attribute
- Identifying Relationship
- Weak Entity
- Partial Key
- Relationship
- Composite Attribute

E-R MODEL DESCRIPTION

The Medical Exigency Database consists of 14 entities in total, out of which 12 are Strong entities, one is a Weak entity, and remaining one is a Relationship.

Strong Entities: Patient, EmergencyLocation, Directory, State, Doctor, Tests, Equipment, Blood, Injection, Medicine, Services, Organ
Weak Entities: Requirements
Relationship Entities: Consults

Assumptions, Cardinality and Participation in Relationships:

1.Patient - BELONGS TO - EmergencyLocation :

The **BelongsTo** relationship describes the relationship between the **patient** and his respective **emergency location**. An affected patient can go to one emergency location only, but each emergency location can hold many patients, hence, there is a **many-to-one** participation between the **Patient** and the **EmergencyLocation**

E-R MODEL DESCRIPTION

2.Patient - CONSULTS - Doctor :

The **CONSULTS** relationship describes the relationship between the patient and the respective consulted doctor. An affected patient can consult one or more doctor, and also each doctor can be consulted by many patients. Hence, there is a **many-to-many** participation between the **Patient** and the **Doctor**. Each patient may or may not consult a doctor, hence there is a **partial** participation from both sides.

3.EmergencyLocation - LOCATED IN - State :

The **LocatedIn** reaction show to which state the EmergencyLocation belongs to. An emergency location always belongs to only one state, while each state can have one ore more emergency locations for the accessibility of the patients located in various parts of the state. Hence, there is a **one-to-many** participation between the EmergencyLocation and the State entity. There is a **partial** participation from EmergencyLocation Side

4.EmergencyLocation-HOLDSCONTACT-Directory-HOLDSCONTACT-Doctor:

The **DIRECTORY** entity is connected with both Emergency Location and Doctor where it holds the contacts of all the Doctors and the EmergencyLocations located in each state, for a convinient communication among the emergency location doctor and the patients. **Here we assume each **doctor** and **emergency location** has only one contact stored in **directory** respectively,hence there is only **one-to-one** relationship here.

E-R MODEL DESCRIPTION

5.Doctor - BELONGS TO - State

The **belongs-to** relation describes the relationship between the **doctor** and the **state** to which they belong. As each state will obviously have more than one Doctor, there will be a **one-to-many** relationship between **State** and **Doctor**. Every state must have a doctor, hence there is a **mandatory** participation of doctors.

6.Patient - HAS - Requirement

The **HAS** relation describes need of the **patient** suggested by the emergency location. **Here we are assuming that every patient's requirement is unique and one patient can have many requirements , hence there is a **one-to-many** relationship between **patient** and the **requirement**. Each patient must have a requirement, hence there is a **mandatory** participation. Note - Requirements is a weak entity

7.Requirement - REQUIRES - Tests

The **NEEDS** relation is in **total-participation** and describes need of a particular requirement. **Here we are assuming that every requirement has one **test** to be performed and the test can be performed by many requirements, hence there is a **one-to-many** relationship between **tests** and the **requirement**. Each test must have a requirement, hence there is a **mandatory** participation.

E-R MODEL DESCRIPTION

8.Requirement - REQUIRES - Equipments

The **REQUIRES** relation is in **total-participation** describes need of a particular requirement. ****Here we are assuming that every requirement has one equipment to be used and the equipment can be utilised by many requirements, hence there is a **one-to-many** relationship between **equipments** and the requirement. Each **equipment** must have a **requirement**, hence there is a **mandatory** participation.

9.Requirement - REQUIRES - Blood

The **REQUIRES** relation is in **total-participation** describes need of a particular requirement. **Here we are assuming that every requirement has a requirement one type of blood which can be utilised by many requirements, hence there is a **one-to-many** relationship between **blood** and the **requirement**. Each equipment must have a requirement, hence there is a **mandatory** participation.

10.Requirement - REQUIRES - Medicine

The **REQUIRES** relation is in **total-participation** describes need of a particular requirement. ****Here we are assuming that every requirement has one medicinal dosage and the medicinal dosage can be performed by many requirements, hence there is a **one-to-many** relationship between **Medicine** and the **requirement**. Each test must have a requirement, hence there is a **mandatory** participation.

E-R MODEL DESCRIPTION

11.Requirement - REQUIRES - Injection

The **REQUIRES** relation is in **total-participation** describes need of a particular **requirement**. **Here we are assuming that every requirement has one Injection to be used and the Injection may be in demand by many requirements, hence there is a **one-to-many** relationship between **Injection** and the **requirement**. Each injection must have a requirement, hence there is a **mandatory** participation.

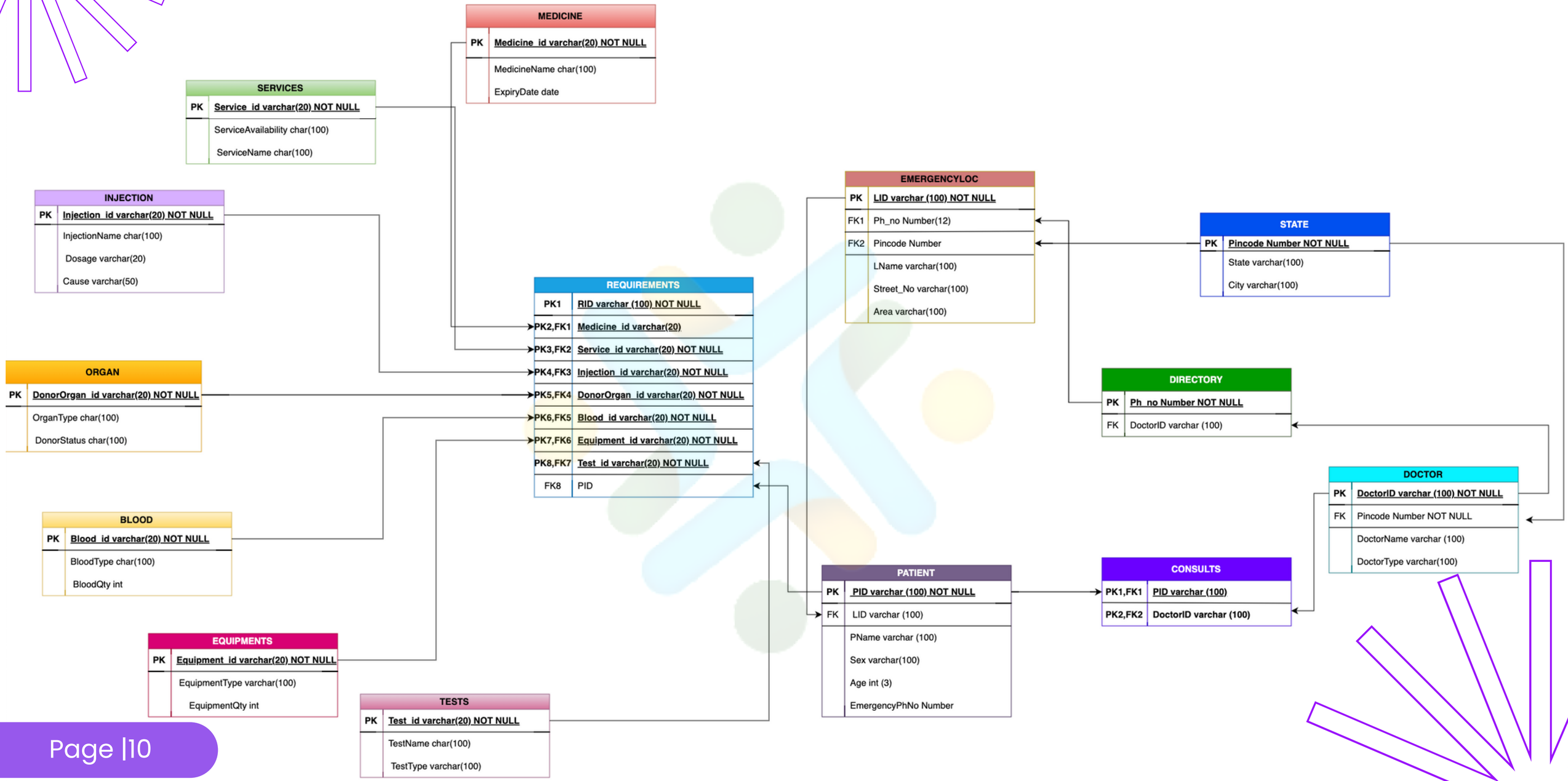
12.Requirement - REQUIRES - Services

The **REQUIRES** relation is in **total-participation** describes need of a particular **requirement**. **Here we are assuming that every requirement has a need one type of service which can be utilized by many requirements, hence there is a **one-to-many** relationship between **Services** and the **requirement**. Each service must have a requirement, hence there is a **mandatory** participation.

13.Requirement - REQUIRES - Organs

The **REQUIRES** relation is in **total-participation** describes need of a particular requirement. **Here we are assuming that every requirement has one organ need and the organ can be in need by many requirements, hence there is a **one-to-many** relationship between **Organs** and the **requirement**. Each organ may or may not be a requirement, hence there is a **partial** participation from both the sides.

RELATIONAL MODEL



FUNCTIONAL DEPENDENCIES AND PRIMARY KEYS

REQUIREMENTS TABLE

- {RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID}->{PID}

- Since PID depends on **rest of all the attributes**,
  (RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID)+ -> R

- Hence, **(RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID)** together is the **PRIMARY KEY**

EMERGENCYLOC TABLE

- {**LID**}->{Ph_no, PinCode, LName, Street_No, Area}

- Since all attributes depends on **LID**,
  (**LID**)+ -> R

- Hence, **LID** is the **PRIMARY KEY** of the EMERGENCYLOC TABLE

FUNCTIONAL DEPENDENCIES AND PRIMARY KEYS

PATIENT TABLE

- {**PID**}->{LID, PName, Sex, Age, EmergencyPhNo}
- Since all attributes are dependent on PID,
  (**PID**)+ -> R
- Hence, **PID** is the **PRIMARY KEY** of the PATIENT TABLE

DIRECTORY TABLE

- {**Ph_no**}->{DoctorID}
- Since DoctorID is dependent on the Ph_no,
  (**Ph_no**)+ -> R
- Hence, **Ph_no** is the **PRIMARY KEY** of the Directory TABLE

FUNCTIONAL DEPENDENCIES AND PRIMARY KEYS

STATE TABLE

- {**PinCode**}->{State, City}
- Since all the attributes are dependent on PinCode,
  (**PinCode**)+ -> R
- Hence, Pincode is the **PRIMARY KEY** of the STATE TABLE

DOCTOR TABLE

- {**DoctorID**} -> {Pincode, DoctorName, DoctorType}
- Since all the attributes are dependent on DoctorId,
  (**DoctorID**)+ -> R
- Hence, **DoctorID** is the **PRIMARY KEY** of the DOCTOR TABLE

FUNCTIONAL DEPENDENCIES AND PRIMARY KEYS

CONSULTS TABLE

- **ASSUMPTION** - In this TABLE either PID or DoctorId can derive the whole relation as both PID and DoctorID are the foreign keys from Patient and Doctor TABLEs having MANY-TO-MANY relationship. We are considering PID to derive all attributes

- {**PID**} -> {DoctorID}

- Since DoctorID is dependent on PID,
  (**PID**)+ -> R

- Hence **PID** is the **PRIMARY KEY** of the **CONSULTS TABLE**

TESTS TABLE

- {**TestID**} -> {TestName, TestType}

- Since all the attributes are dependant on the TestID,
  (**TestID**)+ -> R

- Hence, **TestID** is the **PRIMARY KEY** of the TESTS TABLE

FUNCTIONAL DEPENDENCIES AND PRIMARY KEYS

MEDICINE TABLE

- {**MedicineID**} -> {MedicineName, ExpiryDate}
- Since all the attributes are dependent on the MedicineID, (**MedicineID**)+ -> R
- Hence, **MedicineID** is the **PRIMARY KEY** of the MEDICINE TABLE

SERVICES TABLE

- {**ServiceID**} -> {ServiceAvailability, ServiceName}
- Since all the attributes are dependent on the ServiceID, (**ServiceID**)+ -> R
- Hence, **ServiceID** is the **PRIMARY KEY** for the Services TABLE

FUNCTIONAL DEPENDENCIES AND PRIMARY KEYS

INJECTION TABLE

- {**InjectionID**} -> {InjectionName, Dosage, Cause}
- Since all the attributes are dependent on InjectionID,
  (**InjectionID**)+ -> R
- Hence, **InjectionID** is the **PRIMARY KEY** of the Injection TABLE

BLOOD TABLE

- {**BloodID**} -> {BloodType, BloodQty}
- Since all the attributes are dependent on the BloodID,
  (**BloodID**)+ -> R
- Hence, **BloodID** is the **PRIMARY KEY** of the BLOOD TABLE

FUNCTIONAL DEPENDENCIES AND PRIMARY KEYS

ORGAN TABLE

- {**DonorOrganID**} -> {OrganType, DonorStatus}
- Since all the attributes are dependant on the DonorOrganID, (**DonorOrganID**)+ -> R
- Hence, **DonorOrganID** is the **PRIMARY KEY** of the ORGAN TABLE

EQUIPMENTS TABLE

- {**EquipmentID**} -> {EquipmentType, EquipmentQty}
- Since all the attributes are dependant on the EquipmentID, (**EquipmentID**)+ -> R

Hence, **EquipmentID** is the **PRIMARY KEY** of the EQUIPMENTS TABLE

NORMALISATION

1NF Normal Form

The first normal form is the starting normal form. A relational database is considered to be in 1st Normal form if it does not have any composite or multi-valued attribute.

In the Medical_Exigency database schema, the entity Emergency_loc was observed to have composite attributes branching out from Address attribute namely- STREET_NO. , AREA.

Conversion

- Every Composite attribute must be dissolved into atomic attributes.
- Every multivalued attribute is to be stored into different tuples, with repeating primary primary keys. Hence, multi-valued attribute are included into primary key as well.

Creation

```
CREATE TABLE EmergencyLOC
(
  LID INT,
  LName VARCHAR(20),
  Street_No INT,
  Area VARCHAR(20),
  Ph_no NUMBER(15),
  PinCode NUMBER(10),
  PRIMARY KEY (LID),
  FOREIGN KEY (Ph_no) REFERENCES Directory(Ph_no),
  FOREIGN KEY (PinCode) REFERENCES State(PinCode)
);
```

Solution

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	LID	NUMBER(38,0)	No	(null)	1	(null)
2	LNAME	NUMBER(38,0)	No	(null)	2	(null)
3	STREET_NO	NUMBER(38,0)	No	(null)	3	(null)
4	AREA	NUMBER(38,0)	No	(null)	4	(null)
5	PH_NO	NUMBER(38,0)	No	(null)	5	(null)
6	PINCODE	NUMBER(38,0)	No	(null)	6	(null)

NORMALISATION

2NF Normal Form

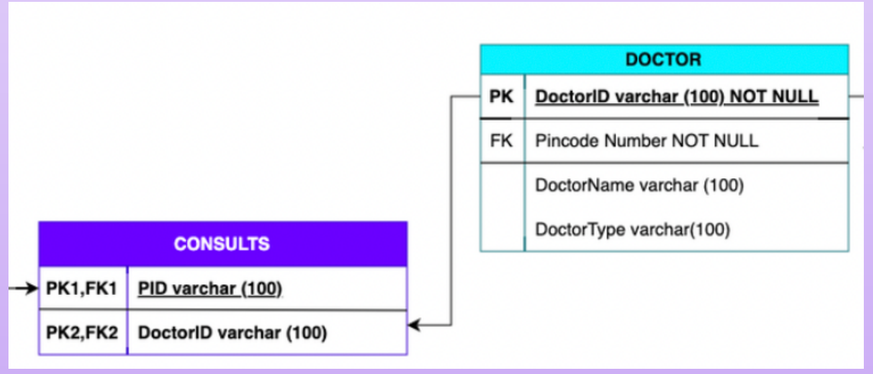
The Second Normal Form (2NF) is based on the concept of full functional dependency. A relational database is considered to be in 2NF if -

- The Relations are in 1NF.
- it has No Partial Dependency, i.e., no non-prime attribute (attributes which are not part of any candidate key) is dependent on any proper subset of any candidate key of the table.
- A relation with a single-attribute primary key is automatically in at least 2NF.
- All Relations in the MEDICAL_EXIGENCY database are in 1NF.
- All Relations have a SINGLE ATTRIBUTE PRIMARY KEY except CONSULTS Entity.

Verification-

- The 'Consults' relational schema is identified by two primary keys namely- Pid(PatientId), DoctorID.
- The DoctorID -> DoctorType, DoctorName but DoctorID itself is a primary key for Doctor Entity. THERFORE ALL RELATIONS in MEDICAL_EXIGENCY Database are in 2NF

Proof-



NORMALISATION

3NF Normal Form 

A relation is in third normal form, if there is no transitive dependency for non-prime attributes as well as it is in second normal form.

- it has No Partial Dependency, i.e., no non-prime attribute (attributes which are not part of any candidate key) is dependent on any proper subset of any candidate key of the table.
- All Relations in the MEDICAL_EXIGENCY database are in 2NF as well as in 1NF respectively.
- Every Relation in the database has a distinctive Primary Key as well as Foreign Key .

Verification-

- Since every relational schema has a unique primary key, this itself ensures that there is no non-prime tending to non-prime attribute. Therefore no partial dependency exists.
- Due to the absence of partial dependency we can quote that - ALL RELATIONS in MEDICAL_EXIGENCY Database are in 3NF

NORMALISATION

BCNF Normal Form

- **BCNF(Boyce Codd Normal Form) in DBMS is an advanced version of 3NF (third normal form). A table or a relation is said to be in BCNF in DBMS if the table or the relation is already in 3NF, and also, for every functional dependency (say, X->Y), X is either the super key or the candidate key.**

Verification- (FD's)

- **Medicine -**
  a. **{MedicineID}** uniquely identifies each record of the Medicine Entity, hence **{MedicineID}** is the **primary key.**
  b. The relation Medicine is in **BCNF**

- **Injection -**
  a. **{InjectionID}** uniquely identifies each record of the Injection Entity, hence **{InjectionID}** is the **primary key.**
  b. The relation Injection is in **BCNF**

Verification- (FD's) contd...

- **Services -**
  a. **{ServiceID}** uniquely identifies each record of the Services Entity, hence **{ServicesID}** is the **primary key.**
  b. The relation Services is in **BCNF**
- **Equipments -**
  a. **{EquipmentID}** uniquely identifies each record of the Equipment Entity, hence **{EquipmentID}** is the **primary key.**
  b. The relation Equipment is in **BCNF**
- **Organs -**
  a. **{OrganID}** uniquely identifies each record of the Organ Entity, hence **{OrganID}** is the **primary key.**
  b. The relation Organ is in **BCNF**
- **Tests-**
  a. **{TestID}** uniquely identifies each record of the Tests Entity, hence **{TestID}** is the **primary key.**
  b. The relation Injection is in **BCNF**

Verification- (FD's) contd...

- **Doctors -**
  a. **{DoctorID}** uniquely identifies each record of the Services Entity, hence **{DoctorID}** is the **primary key.**
  b. The relation Doctors is in **BCNF**
- **Patients -**
  a. **{PatientsID}** uniquely identifies each record of the Patient Entity, hence **{PatientID}** is the **primary key.**
  b. The relation Patient is in **BCNF**
- **State-**
  a. **{Pincode}** uniquely identifies each record of the State Entity, hence **{Pinched}** is the **primary key.**
  b. The relation State is in **BCNF**
- **Emergency Location-**
  a. **{LID}** uniquely identifies each record of the EmergencyLoc Entity, hence **{LID}** is the **primary key.**
  b. The relation EmergencyLoc is in **BCNF**

Verification- (FD's) contd...

- **Requirements -**
  a. **{RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID}** uniquely identifies each record of the Requirements Entity, hence **{RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID}** is the **primary key.**
  b. The relation Requirements is in **BCNF**

- **Directory -**
  a. **{Ph_no}** uniquely identifies each record of the Directory Entity, hence **{Ph_no}** is the **primary key.**
  b. The relation Directory is in **BCNF**

RELATIONAL SCHEMA

REQUIREMENTS TABLE

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	RID	NUMBER(38,0)	No	(null)	1	(null)
2	MEDICINEID	NUMBER(38,0)	No	(null)	2	(null)
3	SERVICEID	NUMBER(38,0)	No	(null)	3	(null)
4	INJECTIONID	NUMBER(38,0)	No	(null)	4	(null)
5	DONORORGANID	NUMBER(38,0)	No	(null)	5	(null)
6	BLOODID	NUMBER(38,0)	No	(null)	6	(null)
7	EQUIPMENTID	NUMBER(38,0)	No	(null)	7	(null)
8	TESTID	NUMBER(38,0)	No	(null)	8	(null)
9	PID	NUMBER(38,0)	No	(null)	9	(null)

RELATIONAL SCHEMA

EMERGENCYLOC TABLE

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	LID	NUMBER(38,0)	No	(null)	1	(null)
2	LNAME	NUMBER(38,0)	No	(null)	2	(null)
3	STREET_NO	NUMBER(38,0)	No	(null)	3	(null)
4	AREA	NUMBER(38,0)	No	(null)	4	(null)
5	PH_NO	NUMBER(38,0)	No	(null)	5	(null)
6	PINCODE	NUMBER(38,0)	No	(null)	6	(null)

PATIENT TABLE

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	PID	NUMBER(38,0)	No	(null)	1	(null)
2	AGE	NUMBER(38,0)	No	(null)	2	(null)
3	PNAME	NUMBER(38,0)	No	(null)	3	(null)
4	SEX	NUMBER(38,0)	No	(null)	4	(null)
5	EMERGENCYPHNO	NUMBER(38,0)	No	(null)	5	(null)
6	LID	NUMBER(38,0)	No	(null)	6	(null)

RELATIONAL SCHEMA

DIRECTORY TABLE

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	PH_NO	NUMBER(38,0)	No	(null)	1	(null)
2	DOCTORID	NUMBER(38,0)	No	(null)	2	(null)

STATE TABLE

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	PINCODE	NUMBER(38,0)	No	(null)	1	(null)
2	STATE	NUMBER(38,0)	No	(null)	2	(null)
3	CITY	NUMBER(38,0)	No	(null)	3	(null)

RELATIONAL SCHEMA

CONSULTS TABLE

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	PID	NUMBER(38,0)	No	(null)	1	(null)
2	DOCTORID	NUMBER(38,0)	No	(null)	2	(null)

DOCTOR TABLE

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	DOCTORID	NUMBER(38,0)	No	(null)	1	(null)
2	DOCTORTYPE	NUMBER(38,0)	No	(null)	2	(null)
3	DOCTORNAME	NUMBER(38,0)	No	(null)	3	(null)
4	PINCODE	NUMBER(38,0)	No	(null)	4	(null)

RELATIONAL SCHEMA

MEDICINE TABLE

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	MEDICINENAME	NUMBER(38,0)	No	(null)	1	(null)
2	MEDICINEID	NUMBER(38,0)	No	(null)	2	(null)
3	EXPIRYDATE	NUMBER(38,0)	No	(null)	3	(null)

SERVICES TABLE

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	SERVICEAVAI...	NUMBER(38,0)	No	(null)	1	(null)
2	SERVICENAME	NUMBER(38,0)	No	(null)	2	(null)
3	SERVICEID	NUMBER(38,0)	No	(null)	3	(null)

RELATIONAL SCHEMA

ORGAN TABLE

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	DONORORGANID	NUMBER(38,0)	No	(null)	1	(null)
2	ORGANTYPE	NUMBER(38,0)	No	(null)	2	(null)
3	DONORSTATUS	NUMBER(38,0)	No	(null)	3	(null)

BLOOD TABLE

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	BLOODID	NUMBER(38,0)	No	(null)	1	(null)
2	BLOODTYPE	NUMBER(38,0)	No	(null)	2	(null)
3	BLOODQTY	NUMBER(38,0)	No	(null)	3	(null)

RELATIONAL SCHEMA

EQUIPMENTS TABLE

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	EQUIPMENTID	NUMBER(38,0)	No	(null)	1	(null)
2	EQUIPMENTTYPE	NUMBER(38,0)	No	(null)	2	(null)
3	EQUIPMENTQTY	NUMBER(38,0)	No	(null)	3	(null)

TESTS TABLE

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	TESTID	NUMBER(38,0)	No	(null)	1	(null)
2	TESTNAME	NUMBER(38,0)	No	(null)	2	(null)
3	TESTTYPE	NUMBER(38,0)	No	(null)	3	(null)

TABLES CREATION

REQUIREMENTS TABLE

```
109 CREATE TABLE Requirements
110 (
111   RID INT NOT NULL,
112   MedicineID INT NOT NULL,
113   ServiceID INT NOT NULL,
114   InjectionID INT NOT NULL,
115   DonorOrganID INT NOT NULL,
116   BloodID INT NOT NULL,
117   EquipmentID INT NOT NULL,
118   TestID INT NOT NULL,
119   PID INT NOT NULL,
120   PRIMARY KEY (RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID),
121   FOREIGN KEY (MedicineID) REFERENCES Medicine(MedicineID),
122   FOREIGN KEY (ServiceID) REFERENCES Services(ServiceID),
123   FOREIGN KEY (InjectionID) REFERENCES Injection(InjectionID),
124   FOREIGN KEY (DonorOrganID) REFERENCES Organ(DonorOrganID),
125   FOREIGN KEY (BloodID) REFERENCES Blood(BloodID),
126   FOREIGN KEY (EquipmentID) REFERENCES Equipments(EquipmentID),
127   FOREIGN KEY (TestID) REFERENCES Tests(TestID),
128   FOREIGN KEY (PID) REFERENCES Patient(PID)
129 );
```

```
Table created.
```

TABLES CREATION

EMERGENCYLOC TABLE

```
84  CREATE TABLE EmergencyLOC
85  (
86    LID INT NOT NULL,
87    LName INT NOT NULL,
88    Street_No INT NOT NULL,
89    Area INT NOT NULL,
90    Ph_no INT NOT NULL,
91    PinCode INT NOT NULL,
92    PRIMARY KEY (LID),
93    FOREIGN KEY (Ph_no) REFERENCES Directory(Ph_no),
94    FOREIGN KEY (PinCode) REFERENCES State(PinCode)
95  );
```

```
Table created.
```

DIRECTORY TABLE

```
76  CREATE TABLE Directory
77  (
78    Ph_no INT NOT NULL,
79    DoctorID INT NOT NULL,
80    PRIMARY KEY (Ph_no),
81    FOREIGN KEY (DoctorID) REFERENCES Doctor(DoctorID)
82  );
```

```
Table created.
```

TABLES CREATION

PATIENT TABLE

```
97  CREATE TABLE Patient
98  (
99    PID INT NOT NULL,
100   Age INT NOT NULL,
101   PName INT NOT NULL,
102   Sex INT NOT NULL,
103   EmergencyPhNo INT NOT NULL,
104   LID INT NOT NULL,
105   PRIMARY KEY (PID),
106   FOREIGN KEY (LID) REFERENCES EmergencyLOC(LID)
107 );
```

```
Table created.
```

STATE TABLE

```
58  CREATE TABLE State
59  (
60    PinCode INT NOT NULL,
61    State INT NOT NULL,
62    City INT NOT NULL,
63    PRIMARY KEY (PinCode)
64  );
```

```
Table created.
```

TABLES CREATION

CONSULTS TABLE

```
131 CREATE TABLE Consults
132 (
133   PID INT NOT NULL,
134   DoctorID INT NOT NULL,
135   PRIMARY KEY (PID, DoctorID),
136   FOREIGN KEY (PID) REFERENCES Patient(PID),
137   FOREIGN KEY (DoctorID) REFERENCES Doctor(DoctorID)
138 );
```

```
Table created.
```

DOCTOR TABLE

```
66 CREATE TABLE Doctor
67 (
68   DoctorID INT NOT NULL,
69   DoctorType INT NOT NULL,
70   DoctorName INT NOT NULL,
71   PinCode INT NOT NULL,
72   PRIMARY KEY (DoctorID),
73   FOREIGN KEY (PinCode) REFERENCES State(PinCode)
74 );
```

```
Table created.
```

TABLES CREATION

MEDICINE TABLE

```
1  CREATE TABLE Medicine
2  (
3    MedicineName INT NOT NULL,
4    MedicineID INT NOT NULL,
5    ExpiryDate INT NOT NULL,
6    PRIMARY KEY (MedicineID)
7  );
```

```
Table created.
```

SERVICES TABLE

```
9  CREATE TABLE Services
10 (
11   ServiceAvailability INT NOT NULL,
12   ServiceName INT NOT NULL,
13   ServiceID INT NOT NULL,
14   PRIMARY KEY (ServiceID)
15 );
```

```
Table created.
```

TABLES CREATION

ORGAN TABLE

```
26  CREATE TABLE Organ
27  (
28    DonorOrganID INT NOT NULL,
29    OrganType INT NOT NULL,
30    DonorStatus INT NOT NULL,
31    PRIMARY KEY (DonorOrganID)
32  );
```

```
Table created.
```

BLOOD TABLE

```
34  CREATE TABLE Blood
35  (
36    BloodID INT NOT NULL,
37    BloodType INT NOT NULL,
38    BloodQty INT NOT NULL,
39    PRIMARY KEY (BloodID)
40  );
```

```
Table created.
```

TABLES CREATION

EQUIPMENTS TABLE

```
42  CREATE TABLE Equipments
43  (
44    EquipmentID INT NOT NULL,
45    EquipmentType INT NOT NULL,
46    EquipmentQty INT NOT NULL,
47    PRIMARY KEY (EquipmentID)
48  );
```

```
Table created.
```

TESTS TABLE

```
50  CREATE TABLE Tests
51  (
52    TestID INT NOT NULL,
53    TestName INT NOT NULL,
54    TestType INT NOT NULL,
55    PRIMARY KEY (TestID)
56  );
```

```
Table created.
```

DATA INSERTION

MEDICINE TABLE

```
INSERT INTO Medicine (MedicineID, MedicineName, ExpiryDate) VALUES (1, 'Paracetamol', '31-DEC-2023');
INSERT INTO Medicine (MedicineID, MedicineName, ExpiryDate) VALUES (2, 'Amoxicillin', '30-SEP-2022');
INSERT INTO Medicine (MedicineID, MedicineName, ExpiryDate) VALUES (3, 'Lisinopril', '15-JUN-2024');
INSERT INTO Medicine (MedicineID, MedicineName, ExpiryDate) VALUES (4, 'Atorvastatin', '31-MAY-2023');
INSERT INTO Medicine (MedicineID, MedicineName, ExpiryDate) VALUES (5, 'Omeprazole', '30-NOV-2024');
INSERT INTO Medicine (MedicineID, MedicineName, ExpiryDate) VALUES (6, 'Metformin', '31-AUG-2023');
INSERT INTO Medicine (MedicineID, MedicineName, ExpiryDate) VALUES (7, 'Sertraline', '28-FEB-2024');
INSERT INTO Medicine (MedicineID, MedicineName, ExpiryDate) VALUES (8, 'Simvastatin', '15-OCT-2023');
INSERT INTO Medicine (MedicineID, MedicineName, ExpiryDate) VALUES (9, 'Losartan', '30-SEP-2023');
INSERT INTO Medicine (MedicineID, MedicineName, ExpiryDate) VALUES (10, 'Levothyroxine', '31-JAN-2024');
SELECT * FROM MEDICINE;
```

	MEDICINENAME	MEDICINEID	EXPIRYDATE
1	Paracetamol	1	31-12-23
2	Amoxicillin	2	30-09-22
3	Lisinopril	3	15-06-24
4	Atorvastatin	4	31-05-23
5	Omeprazole	5	30-11-24
6	Metformin	6	31-08-23
7	Sertraline	7	28-02-24
8	Simvastatin	8	15-10-23
9	Losartan	9	30-09-23
10	Levothyroxine	10	31-01-24

ORGAN TABLE

```
INSERT INTO Organ (DonorOrganID, OrganType, DonorStatus) VALUES (21, 'Heart', 'Available');
INSERT INTO Organ (DonorOrganID, OrganType, DonorStatus) VALUES (22, 'Kidney', 'Unavailable');
INSERT INTO Organ (DonorOrganID, OrganType, DonorStatus) VALUES (23, 'Liver', 'Available');
INSERT INTO Organ (DonorOrganID, OrganType, DonorStatus) VALUES (24, 'Lung', 'Available');
INSERT INTO Organ (DonorOrganID, OrganType, DonorStatus) VALUES (25, 'Pancreas', 'Unavailable');
INSERT INTO Organ (DonorOrganID, OrganType, DonorStatus) VALUES (26, 'Cornea', 'Available');
INSERT INTO Organ (DonorOrganID, OrganType, DonorStatus) VALUES (27, 'Bone Marrow', 'Unavailable');
INSERT INTO Organ (DonorOrganID, OrganType, DonorStatus) VALUES (28, 'Intestine', 'Available');
INSERT INTO Organ (DonorOrganID, OrganType, DonorStatus) VALUES (29, 'Skin', 'Available');
INSERT INTO Organ (DonorOrganID, OrganType, DonorStatus) VALUES (290, 'Heart', 'Unavailable');
```

	DONORORGANID	ORGANTYPE	DONORSTATUS
1	21	Heart	Available
2	22	Kidney	Unavailable
3	23	Liver	Available
4	24	Lung	Available
5	25	Pancreas	Unavailable
6	26	Cornea	Available
7	27	Bone Marrow	Unavailable
8	28	Intestine	Available
9	29	Skin	Available
10	290	Heart	Unavailable

DATA INSERTION

INJECTION TABLE

```
INSERT INTO Injection (InjectionName, Dosage, Cause, InjectionID) VALUES ('Flu Vaccine', 50, 'Prevention of influenza', 1010);
INSERT INTO Injection (InjectionName, Dosage, Cause, InjectionID) VALUES ('Tetanus Toxoid', 10, 'Protection against tetanus', 20);
INSERT INTO Injection (InjectionName, Dosage, Cause, InjectionID) VALUES ('Hepatitis B Vaccine', 20, 'Prevention of hepatitis B', 30);
INSERT INTO Injection (InjectionName, Dosage, Cause, InjectionID) VALUES ('Insulin', 5, 'Diabetes management', 40);
INSERT INTO Injection (InjectionName, Dosage, Cause, InjectionID) VALUES ('Penicillin', 250, 'Treatment of bacterial infections', 50);
INSERT INTO Injection (InjectionName, Dosage, Cause, InjectionID) VALUES ('Epinephrine', 0.3, 'Emergency treatment of severe allergic reactions', 60);
INSERT INTO Injection (InjectionName, Dosage, Cause, InjectionID) VALUES ('Vitamin B12 Injection', 1000, 'Supplementation for vitamin B12 deficiency', 70);
INSERT INTO Injection (InjectionName, Dosage, Cause, InjectionID) VALUES ('Depo-Provera', 150, 'Long-acting birth control', 80);
INSERT INTO Injection (InjectionName, Dosage, Cause, InjectionID) VALUES ('Intravenous Immunoglobulin', 400, 'Immune system disorders', 90);
INSERT INTO Injection (InjectionName, Dosage, Cause, InjectionID) VALUES ('Morphine', 10, 'Pain management', 100);
```

	INJECTIONNAME	DOSAGE	CAUSE	INJECTIONID
1	Flu Vaccine	50	Prevention of influenza	1010
2	Tetanus Toxoid	10	Protection against tetanus	20
3	Hepatitis B Vaccine	20	Prevention of hepatitis B	30
4	Insulin	5	Diabetes management	40
5	Penicillin	250	Treatment of bacterial infections	50
6	Epinephrine	0	Emergency treatment of severe allergic reactions	60
7	Vitamin B12 Injection	1000	Supplementation for vitamin B12 deficiency	70
8	Depo-Provera	150	Long-acting birth control	80
9	Intravenous Immunoglobulin	400	Immune system disorders	90
10	Morphine	10	Pain management	100

DATA INSERTION

SERVICES TABLE

```
INSERT INTO Services (ServiceAvailability, ServiceName, ServiceID) VALUES ('Available', 'Ambulance', 101);
INSERT INTO Services (ServiceAvailability, ServiceName, ServiceID) VALUES ('Available', 'FirstAid', 102);
INSERT INTO Services (ServiceAvailability, ServiceName, ServiceID) VALUES ('Unavailable', 'Poison_Control', 103);
INSERT INTO Services (ServiceAvailability, ServiceName, ServiceID) VALUES ('Available', 'Paramedics', 104);
INSERT INTO Services (ServiceAvailability, ServiceName, ServiceID) VALUES ('Available', 'Burn_Unit', 105);
INSERT INTO Services (ServiceAvailability, ServiceName, ServiceID) VALUES ('Available', 'Rehabilitation', 106);
INSERT INTO Services (ServiceAvailability, ServiceName, ServiceID) VALUES ('Available', 'Nutritionist', 107);
INSERT INTO Services (ServiceAvailability, ServiceName, ServiceID) VALUES ('Unavailable', 'Airlift', 108);
INSERT INTO Services (ServiceAvailability, ServiceName, ServiceID) VALUES ('Available', 'Palliative_Service', 109);
INSERT INTO Services (ServiceAvailability, ServiceName, ServiceID) VALUES ('Available', 'Nursing', 110);
```

SERVICEAVAILABILITY	SERVICENAME	SERVICEID
Available	Ambulance	101
Available	FirstAid	102
Unavailable	Poison_Control	103
Available	Paramedics	104
Available	Burn_Unit	105
Available	Rehabilitation	106
Available	Nutritionist	107
Unavailable	Airlift	108
Available	Palliative_Service	109
Available	Nursing	110

10 rows selected.

DATA INSERTION

BLOOD TABLE

```
INSERT INTO Blood (BloodID, BloodType, BloodQty) VALUES (41, 'A+', 100);
INSERT INTO Blood (BloodID, BloodType, BloodQty) VALUES(42, 'B+', 75);
INSERT INTO Blood (BloodID, BloodType, BloodQty) VALUES(43, 'O+', 150);
INSERT INTO Blood (BloodID, BloodType, BloodQty) VALUES(44, 'AB+', 50);
INSERT INTO Blood (BloodID, BloodType, BloodQty) VALUES(45, 'A-', 30);
INSERT INTO Blood (BloodID, BloodType, BloodQty) VALUES(46, 'B-', 20);
INSERT INTO Blood (BloodID, BloodType, BloodQty) VALUES(47, 'O-', 80);
INSERT INTO Blood (BloodID, BloodType, BloodQty) VALUES(48, 'AB-', 10);
INSERT INTO Blood (BloodID, BloodType, BloodQty) VALUES(49, 'A+', 150);
INSERT INTO Blood (BloodID, BloodType, BloodQty) VALUES(490, 'O+', 150);
SELECT * FROM BLOOD;
```

BLOODID	BLOODTYPE	BLOODQTY
41	A+	100
42	B+	75
43	O+	150
44	AB+	50
45	A-	30
46	B-	20
47	O-	80
48	AB-	10
49	A+	150
490	O+	150

10 rows selected.

EQUIPMENT TABLE

```
INSERT INTO Equipments (EquipmentID, EquipmentType, EquipmentQty) VALUES (71, 'Stethoscope', 5);
INSERT INTO Equipments (EquipmentID, EquipmentType, EquipmentQty) VALUES (72, 'Surgical Scissors', 10);
INSERT INTO Equipments (EquipmentID, EquipmentType, EquipmentQty) VALUES (73, 'Blood Pressure Monitor', 3);
INSERT INTO Equipments (EquipmentID, EquipmentType, EquipmentQty) VALUES (74, 'Defibrillator', 2);
INSERT INTO Equipments (EquipmentID, EquipmentType, EquipmentQty) VALUES (75, 'X-Ray Machine', 1);
INSERT INTO Equipments (EquipmentID, EquipmentType, EquipmentQty) VALUES (76, 'Surgical Gloves', 50);
INSERT INTO Equipments (EquipmentID, EquipmentType, EquipmentQty) VALUES (77, 'Infusion Pump', 4);
INSERT INTO Equipments (EquipmentID, EquipmentType, EquipmentQty) VALUES (78, 'Oxygen Concentrator', 2);
INSERT INTO Equipments (EquipmentID, EquipmentType, EquipmentQty) VALUES (79, 'Surgical Mask', 100);
INSERT INTO Equipments (EquipmentID, EquipmentType, EquipmentQty) VALUES (790, 'Autoclave', 2);
```

	EQUIPMENTID	EQUIPMENTTYPE	EQUIPMENTQTY
1	71	Stethoscope	5
2	72	Surgical Scissors	10
3	73	Blood Pressure Monitor	3
4	74	Defibrillator	2
5	75	X-Ray Machine	1
6	76	Surgical Gloves	50
7	77	Infusion Pump	4
8	78	Oxygen Concentrator	2
9	79	Surgical Mask	100
10	790	Autoclave	2

DATA INSERTION

TESTS TABLE

```
INSERT INTO Tests (TestID, TestName, TestType) VALUES (91, 'Complete Blood Count', 'Blood Test');
INSERT INTO Tests (TestID, TestName, TestType) VALUES (92, 'X-Ray', 'Radiology');
INSERT INTO Tests (TestID, TestName, TestType) VALUES (93, 'Electrocardiogram', 'Cardiology');
INSERT INTO Tests (TestID, TestName, TestType) VALUES (94, 'Urinalysis', 'Laboratory');
INSERT INTO Tests (TestID, TestName, TestType) VALUES (95, 'MRI Scan', 'Radiology');
INSERT INTO Tests (TestID, TestName, TestType) VALUES (96, 'Biopsy', 'Pathology');
INSERT INTO Tests (TestID, TestName, TestType) VALUES (97, 'Colonoscopy', 'Gastroenterology');
INSERT INTO Tests (TestID, TestName, TestType) VALUES (98, 'Pulmonary Function Test', 'Respiratory');
INSERT INTO Tests (TestID, TestName, TestType) VALUES (99, 'Allergy Testing', 'Immunology');
INSERT INTO Tests (TestID, TestName, TestType) VALUES (990, 'Echocardiogram', 'Cardiology');
```

	TESTID	TESTNAME	TESTTYPE
1	91	Complete Blood Count	Blood Test
2	92	X-Ray	Radiology
3	93	Electrocardiogram	Cardiology
4	94	Urinalysis	Laboratory
5	95	MRI Scan	Radiology
6	96	Biopsy	Pathology
7	97	Colonoscopy	Gastroenterology
8	98	Pulmonary Function Test	Respiratory
9	99	Allergy Testing	Immunology
10	990	Echocardiogram	Cardiology

STATE TABLE

```
INSERT INTO State (PinCode, State, City) VALUES (110001, 'Delhi', 'New Delhi');
INSERT INTO State (PinCode, State, City) VALUES (400001, 'Maharashtra', 'Mumbai');
INSERT INTO State (PinCode, State, City) VALUES (560001, 'Karnataka', 'Bengaluru');
INSERT INTO State (PinCode, State, City) VALUES (600001, 'Tamil Nadu', 'Chennai');
INSERT INTO State (PinCode, State, City) VALUES (700001, 'West Bengal', 'Kolkata');
INSERT INTO State (PinCode, State, City) VALUES (500001, 'Telangana', 'Hyderabad');
INSERT INTO State (PinCode, State, City) VALUES (380001, 'Gujarat', 'Ahmedabad');
INSERT INTO State (PinCode, State, City) VALUES (110011, 'Uttar_Pradesh', 'Lucknow');
INSERT INTO State (PinCode, State, City) VALUES (641001, 'Rajasthan', 'Jaipur');
INSERT INTO State (PinCode, State, City) VALUES (110021, 'Delhi', 'Ghaziabad');
```

	PINCODE	STATE	CITY
1	110001	Delhi	New Delhi
2	400001	Maharashtra	Mumbai
3	560001	Karnataka	Bengaluru
4	600001	Tamil Nadu	Chennai
5	700001	West Bengal	Kolkata
6	500001	Telangana	Hyderabad
7	380001	Gujarat	Ahmedabad
8	110011	Uttar_Pradesh	Lucknow
9	641001	Rajasthan	Jaipur
10	110021	Delhi	Ghaziabad

DATA INSERTION

DOCTOR TABLE

```
INSERT INTO Doctor (DoctorID, DoctorType, DoctorName, PinCode) VALUES (51, 'Cardiologist', 'Dr. John Smith', 110001);
INSERT INTO Doctor (DoctorID, DoctorType, DoctorName, PinCode) VALUES (52, 'Dermatologist', 'Dr. Sarah Johnson', 400001);
INSERT INTO Doctor (DoctorID, DoctorType, DoctorName, PinCode) VALUES (53, 'Orthopedic Surgeon', 'Dr. Michael Brown', 560001);
INSERT INTO Doctor (DoctorID, DoctorType, DoctorName, PinCode) VALUES (54, 'Gynecologist', 'Dr. Emily Davis', 600001);
INSERT INTO Doctor (DoctorID, DoctorType, DoctorName, PinCode) VALUES (55, 'Pediatrician', 'Dr. Robert Wilson', 700001);
INSERT INTO Doctor (DoctorID, DoctorType, DoctorName, PinCode) VALUES (56, 'Neurologist', 'Dr. Jessica Lee', 500001);
INSERT INTO Doctor (DoctorID, DoctorType, DoctorName, PinCode) VALUES (57, 'Ophthalmologist', 'Dr. David Miller', 380001);
INSERT INTO Doctor (DoctorID, DoctorType, DoctorName, PinCode) VALUES (58, 'Urologist', 'Dr. Jennifer Thompson', 110011);
INSERT INTO Doctor (DoctorID, DoctorType, DoctorName, PinCode) VALUES (59, 'Endocrinologist', 'Dr. Christopher Clark', 641001);
INSERT INTO Doctor (DoctorID, DoctorType, DoctorName, PinCode) VALUES (590, 'Psychiatrist', 'Dr. Elizabeth White', 110021);
```

	DOCTORID	DOCTORTYPE	DOCTORNAME	PINCODE
1	51	Cardiologist	Dr. John Smith	110001
2	52	Dermatologist	Dr. Sarah Johnson	400001
3	53	Orthopedic Surgeon	Dr. Michael Brown	560001
4	54	Gynecologist	Dr. Emily Davis	600001
5	55	Pediatrician	Dr. Robert Wilson	700001
6	56	Neurologist	Dr. Jessica Lee	500001
7	57	Ophthalmologist	Dr. David Miller	380001
8	58	Urologist	Dr. Jennifer Thompson	110011
9	59	Endocrinologist	Dr. Christopher Clark	641001
10	590	Psychiatrist	Dr. Elizabeth White	110021

DATA INSERTION

DIRECTORY TABLE

```
INSERT INTO Directory (Ph_no, DoctorID) VALUES (1234567890, 51);
INSERT INTO Directory (Ph_no, DoctorID) VALUES (9876543210, 52);
INSERT INTO Directory (Ph_no, DoctorID) VALUES (2345678901, 53);
INSERT INTO Directory (Ph_no, DoctorID) VALUES (8901234567, 54);
INSERT INTO Directory (Ph_no, DoctorID) VALUES (4567890123, 55);
INSERT INTO Directory (Ph_no, DoctorID) VALUES (7890123456, 56);
INSERT INTO Directory (Ph_no, DoctorID) VALUES (3456789012, 57);
INSERT INTO Directory (Ph_no, DoctorID) VALUES (9012345678, 58);
INSERT INTO Directory (Ph_no, DoctorID) VALUES (5678901234, 59);
INSERT INTO Directory (Ph_no, DoctorID) VALUES (0123456789, 590);
```

	PH_NO	DOCTORID
1	1234567890	51
2	9876543210	52
3	2345678901	53
4	8901234567	54
5	4567890123	55
6	7890123456	56
7	3456789012	57
8	9012345678	58
9	5678901234	59
10	123456789	590

CONSULTS TABLE

```
INSERT INTO Consults (PID, DoctorID) VALUES (61, 51);
INSERT INTO Consults (PID, DoctorID) VALUES (62, 52);
INSERT INTO Consults (PID, DoctorID) VALUES (63, 53);
INSERT INTO Consults (PID, DoctorID) VALUES (64, 54);
INSERT INTO Consults (PID, DoctorID) VALUES (65, 55);
INSERT INTO Consults (PID, DoctorID) VALUES (66, 56);
INSERT INTO Consults (PID, DoctorID) VALUES (67, 57);
INSERT INTO Consults (PID, DoctorID) VALUES (68, 58);
INSERT INTO Consults (PID, DoctorID) VALUES (69, 59);
INSERT INTO Consults (PID, DoctorID) VALUES (690, 590);
```

	PID	DOCTORID
1	61	51
2	62	52
3	63	53
4	64	54
5	65	55
6	66	56
7	67	57
8	68	58
9	69	59
10	690	590

DATA INSERTION

EMERGENCYLOC TABLE

```
INSERT INTO EmergencyLOC (LID, LName, Street_No, Area, Ph_no, PinCode) VALUES (31, 'Emergency Hospital', 123, 'Main Street', 1234567890, 110001);
INSERT INTO EmergencyLOC (LID, LName, Street_No, Area, Ph_no, PinCode) VALUES (32, 'City Medical Center', 456, 'Park Avenue', 9876543210, 400001);
INSERT INTO EmergencyLOC (LID, LName, Street_No, Area, Ph_no, PinCode) VALUES (33, 'Healthcare Clinic', 789, 'Lake Road', 2345678901, 560001);
INSERT INTO EmergencyLOC (LID, LName, Street_No, Area, Ph_no, PinCode) VALUES (34, 'Urgent Care Center', 321, 'Garden Street', 8901234567, 600001);
INSERT INTO EmergencyLOC (LID, LName, Street_No, Area, Ph_no, PinCode) VALUES (35, 'Medical Services', 654, 'River View Road', 4567890123, 700001);
INSERT INTO EmergencyLOC (LID, LName, Street_No, Area, Ph_no, PinCode) VALUES (36, 'Emergency Care', 987, 'Hillside Avenue', 7890123456, 500001);
INSERT INTO EmergencyLOC (LID, LName, Street_No, Area, Ph_no, PinCode) VALUES (37, 'Health Center', 741, 'Ocean Drive', 3456789012, 380001);
INSERT INTO EmergencyLOC (LID, LName, Street_No, Area, Ph_no, PinCode) VALUES (38, 'Critical Care', 852, 'Sunset Boulevard', 9012345678, 110011);
INSERT INTO EmergencyLOC (LID, LName, Street_No, Area, Ph_no, PinCode) VALUES (39, 'Emergency Clinic', 963, 'Skyline Street', 5678901234, 641001);
INSERT INTO EmergencyLOC (LID, LName, Street_No, Area, Ph_no, PinCode) VALUES (390, 'Health Services', 159, 'Meadow Lane', 0123456789, 110021);
```

	LID	LNAME	STREET_NO	AREA	PH_NO	PINCODE
1	31	Emergency Hospital	123	Main Street	1234567890	110001
2	32	City Medical Center	456	Park Avenue	9876543210	400001
3	33	Healthcare Clinic	789	Lake Road	2345678901	560001
4	34	Urgent Care Center	321	Garden Street	8901234567	600001
5	35	Medical Services	654	River View Road	4567890123	700001
6	36	Emergency Care	987	Hillside Avenue	7890123456	500001
7	37	Health Center	741	Ocean Drive	3456789012	380001
8	38	Critical Care	852	Sunset Boulevard	9012345678	110011
9	39	Emergency Clinic	963	Skyline Street	5678901234	641001
10	390	Health Services	159	Meadow Lane	123456789	110021

DATA INSERTION

PATIENT TABLE

```
INSERT INTO Patient (PID, Age, PName, Sex, EmergencyPhNo, LID) VALUES (61, 25, 'John Doe', 'Male', 1234567890, 31);
INSERT INTO Patient (PID, Age, PName, Sex, EmergencyPhNo, LID) VALUES (62, 30, 'Jane Smith', 'Female', 9876543210, 32);
INSERT INTO Patient (PID, Age, PName, Sex, EmergencyPhNo, LID) VALUES (63, 45, 'Michael Johnson', 'Male', 2345678901, 33);
INSERT INTO Patient (PID, Age, PName, Sex, EmergencyPhNo, LID) VALUES (64, 60, 'Sarah Davis', 'Female', 8901234567, 34);
INSERT INTO Patient (PID, Age, PName, Sex, EmergencyPhNo, LID) VALUES (65, 35, 'David Wilson', 'Male', 4567890123, 35);
INSERT INTO Patient (PID, Age, PName, Sex, EmergencyPhNo, LID) VALUES (66, 28, 'Emily Thompson', 'Female', 7890123456, 36);
INSERT INTO Patient (PID, Age, PName, Sex, EmergencyPhNo, LID) VALUES (67, 42, 'Robert Miller', 'Male', 3456789012, 37);
INSERT INTO Patient (PID, Age, PName, Sex, EmergencyPhNo, LID) VALUES (68, 50, 'Jennifer Clark', 'Female', 9012345678, 38);
INSERT INTO Patient (PID, Age, PName, Sex, EmergencyPhNo, LID) VALUES (69, 38, 'Daniel Brown', 'Male', 5678901234, 39);
INSERT INTO Patient (PID, Age, PName, Sex, EmergencyPhNo, LID) VALUES (690, 32, 'Jessica Lee', 'Female', 0123456789, 390);
```

	PID	AGE	PNAME	SEX	EMERGENCYPHNO	LID
1	61	25	John Doe	Male	1234567890	31
2	62	30	Jane Smith	Female	9876543210	32
3	63	45	Michael Johnson	Male	2345678901	33
4	64	60	Sarah Davis	Female	8901234567	34
5	65	35	David Wilson	Male	4567890123	35
6	66	28	Emily Thompson	Female	7890123456	36
7	67	42	Robert Miller	Male	3456789012	37
8	68	50	Jennifer Clark	Female	9012345678	38
9	69	38	Daniel Brown	Male	5678901234	39
10	690	32	Jessica Lee	Female	123456789	390

DATA INSERTION

REQUIREMENT TABLE

```
INSERT INTO Requirements (RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID, PID) VALUES (11, 1, 101, 1010, 21, 41, 71, 91, 61);
INSERT INTO Requirements (RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID, PID) VALUES (12, 2, 102, 20, 22, 42, 72, 92, 62);
INSERT INTO Requirements (RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID, PID) VALUES (13, 3, 103, 30, 23, 43, 73, 93, 63);
INSERT INTO Requirements (RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID, PID) VALUES (14, 4, 104, 40, 24, 44, 74, 94, 64);
INSERT INTO Requirements (RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID, PID) VALUES (15, 5, 105, 50, 25, 45, 75, 95, 65);
INSERT INTO Requirements (RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID, PID) VALUES (16, 6, 106, 60, 26, 46, 76, 96, 66);
INSERT INTO Requirements (RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID, PID) VALUES (17, 7, 107, 70, 27, 47, 77, 97, 67);
INSERT INTO Requirements (RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID, PID) VALUES (18, 8, 108, 80, 28, 48, 78, 98, 68);
INSERT INTO Requirements (RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID, PID) VALUES (19, 9, 109, 90, 29, 49, 79, 99,69);
INSERT INTO Requirements (RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID, PID) VALUES (190, 10, 110, 100, 290, 490, 790, 990, 690);
INSERT INTO Requirements (RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID, PID) VALUES (12, 8, 104, 90, 26, 44, 73, 93, 69);
INSERT INTO Requirements (RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID, PID) VALUES (14, 7, 103, 40, 22, 49, 75, 93, 68);
INSERT INTO Requirements (RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID, PID) VALUES (16, 8, 104, 40, 21, 43, 71, 990, 690);
INSERT INTO Requirements (RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID, PID) VALUES (19, 9, 101, 80, 28, 45, 73, 97, 62);
INSERT INTO Requirements (RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID, PID) VALUES (14, 8, 107, 60, 22, 49, 74, 96, 61);
INSERT INTO Requirements (RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID, PID) VALUES (14, 2, 109, 20, 28, 49, 79, 97, 63);
INSERT INTO Requirements (RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID, PID) VALUES (13, 9, 106, 70, 290, 48, 790, 96, 61);
INSERT INTO Requirements (RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID, PID) VALUES (11, 8, 106, 70, 290, 48, 78, 96, 61);
INSERT INTO Requirements (RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID, PID) VALUES (11, 8, 106, 70, 290, 48, 78, 96, 61);
INSERT INTO Requirements (RID, MedicineID, ServiceID, InjectionID, DonorOrganID, BloodID, EquipmentID, TestID, PID) VALUES (11, 8, 106, 70, 290, 48, 78, 96, 61);
```

DATA INSERTION

REQUIREMENT TABLE

RID	MEDICINEID	SERVICEID	INJECTIONID	DONORORGANID	BLOODID	EQUIPMENTID	TESTID	PID
11	8	106	70	290	48	78	96	61
18	6	109	50	27	41	71	93	65
14	1	101	50	29	490	78	99	69
11	1	101	1010	21	41	71	91	61
12	2	102	20	22	42	72	92	62
13	3	103	30	23	43	73	93	63
14	4	104	40	24	44	74	94	64
15	5	105	50	25	45	75	95	65
16	6	106	60	26	46	76	96	66
17	7	107	70	27	47	77	97	67
18	8	108	80	28	48	78	98	68
19	9	109	90	29	49	79	99	69
190	10	110	100	290	490	790	990	690
14	2	109	20	28	49	79	97	63
13	9	106	70	290	48	790	96	61
12	8	104	90	26	44	73	93	69
14	7	103	40	22	49	75	93	68
19	9	101	80	28	45	73	97	62
14	8	107	60	22	49	74	96	61
16	8	104	40	21	43	71	990	690

SQL - QUERIES

QUERY - 1

Retrieve all the medicines that will expire by the end of this year.

```
1  --1. Retrieve all the medicines that will expire by the end of this year.
2  SELECT * FROM MEDICINE WHERE EXTRACT(YEAR FROM ExpiryDate)=EXTRACT(YEAR FROM SYSDATE);
```

MEDICINENAME	MEDICINEID	EXPIRYDATE
Paracetamol	1	31-DEC-23
Atorvastatin	4	31-MAY-23
Metformin	6	31-AUG-23
Simvastatin	8	15-OCT-23
Losartan	9	30-SEP-23

SQL – QUERIES

QUERY - 2

Retrieve all the emergency locations and their contact details in Hyderabad.

```
1  SELECT LName, Street_No, Area, Ph_no FROM EmergencyLOC
2  INNER JOIN State ON EmergencyLOC.PinCode = State.PinCode
3  WHERE State.City = 'Hyderabad';
```

LNAME	STREET_NO	AREA	PH_NO
Emergency Care	987	Hillside Avenue	7890123456

QUERY - 3

Retrieve the total quantity of blood of a 'AB+' type available in the inventory.

```
1  --3. Retrieve the total quantity of blood of a 'AB+' type available in the inventory.
2  SELECT BloodType, SUM(BloodQty) as TOTAL FROM Blood GROUP BY BloodType HAVING BloodType = 'AB+';
```

BLOODTYPE	TOTAL
AB+	50

SQL – QUERIES

QUERY - 4

Retrieve the list of all doctors and their phone numbers in Delhi.

```
1  --4. Retrieve the list of all doctors and their phone numbers in Delhi.
2  SELECT DoctorName, Directory.Ph_no FROM Doctor
3  INNER JOIN Directory ON Doctor.DoctorID = Directory.DoctorID
4  INNER JOIN State ON Doctor.PinCode = State.PinCode
5  WHERE State.State = 'Delhi';
```

DOCTORNAME	PH_NO
Dr. John Smith	1234567890
Dr. Elizabeth White	123456789

QUERY - 5

Retrieve the list of all patients who are waiting for Liver donation.

```
1  --5. Retrieve the list of all patients who are waiting for Liver donation.
2  SELECT Patient.PName, Organ.OrganType FROM Patient
3  INNER JOIN Requirements ON Patient.PID = Requirements.PID
4  INNER JOIN Organ ON Requirements.DonorOrganID = Organ.DonorOrganID
5  WHERE Organ.OrganType = 'Liver';
```

PNAME	ORGANTYPE
Michael Johnson	Liver

SQL - QUERIES

QUERY - 6

Retrieve the list of all tests and their types.

```sql
--6. Retrieve the list of all tests and their types.
SELECT TestName, TestType FROM Tests;
```

TESTNAME	TESTTYPE
Complete Blood Count	Blood Test
X-Ray	Radiology
Electrocardiogram	Cardiology
Urinalysis	Laboratory
MRI Scan	Radiology
Biopsy	Pathology
Colonoscopy	Gastroenterology
Pulmonary Function Test	Respiratory
Allergy Testing	Immunology
Echocardiogram	Cardiology

QUERY - 7

Retrieve the list of all injections for Supplementation for vitamin B12 deficiency.

```sql
--7. Retrieve the list of all injections for Supplementation for vitamin B12 deficiency.
SELECT InjectionName, Dosage FROM Injection WHERE Cause = 'Supplementation for vitamin B12 deficiency';
```

INJECTIONNAME	DOSAGE
Vitamin B12 Injection	1000

SQL - QUERIES

QUERY - 8

Retrieve the list of all emergency locations in a pin code - 560001.

```
1  --8. Retrieve the list of all emergency locations in a pin code - 560001.
2  SELECT LName, Street_No, Area, Ph_no FROM EmergencyLOC
3  INNER JOIN State ON EmergencyLOC.PinCode = State.PinCode
4  WHERE State.PinCode = '560001';
```

LNAME	STREET_NO	AREA	PH_NO
Healthcare Clinic	789	Lake Road	2345678901

SQL – QUERIES

QUERY - 9

Retrieve the list of all patients and their emergency contact details.

```
1  --9. Retrieve the list of all patients and their emergency contact details.
2  SELECT PName, EmergencyPhNo FROM Patient;
```

PNAME	EMERGENCYPHNO
John Doe	1234567890
Jane Smith	9876543210
Michael Johnson	2345678901
Sarah Davis	8901234567
David Wilson	4567890123
Emily Thompson	7890123456
Robert Miller	3456789012
Jennifer Clark	9012345678
Daniel Brown	5678901234
Jessica Lee	123456789

SQL – QUERIES

QUERY - 10

Retrieve the list of all equipment types and their quantities in the inventory.

```
1  --10. Retrieve the list of all equipment types and their quantities in the inventory.
2  SELECT EquipmentType, EquipmentQty FROM Equipments;
```

EQUIPMENTTYPE	EQUIPMENTQTY
Stethoscope	5
Surgical Scissors	10
Blood Pressure Monitor	3
Defibrillator	2
X-Ray Machine	1
Surgical Gloves	50
Infusion Pump	4
Oxygen Concentrator	2
Surgical Mask	100
Autoclave	2

TRIGGERS & PROCEDURES

Create a row level trigger for the BLOOD TABLE that would fire for INSERT OR UPDATE OR DELETE Operations performed on the BLOOD TABLE. This trigger will display the quantity difference between the old values and new values:

```
CREATE OR REPLACE TRIGGER QUANTITY_CHANGE
BEFORE DELETE OR INSERT OR UPDATE ON Blood
FOR EACH ROW
WHEN (NEW.BLOODID>0)
DECLARE
quan_diff number;
BEGIN
quan_diff := :NEW.BloodQty - :OLD.BloodQty;
dbms_output.put_line('OLD QUANTITY:' || :OLD.BloodQty);
dbms_output.put_line('NEW QUANTITY:' || :NEW.BloodQty);
dbms_output.put_line('QUANTITY DIFFERENCE:' || quan_diff);
END;
/

DECLARE
total_rows number(2);
BEGIN
UPDATE BLOOD
SET BloodQty= BloodQty + 50;
IF sql%notfound THEN
dbms_output.put_line('quantity not updated');
ELSE
total_rows:= sql%rowcount;
dbms_output.put_line(total_rows || 'Quantity Updated' );
END IF;
END;
/
```

```
Trigger created.


Statement processed.
OLD QUANTITY:150
NEW QUANTITY:200
QUANTITY DIFFERENCE:50
OLD QUANTITY:125
NEW QUANTITY:175
QUANTITY DIFFERENCE:50
OLD QUANTITY:200
NEW QUANTITY:250
QUANTITY DIFFERENCE:50
OLD QUANTITY:100
NEW QUANTITY:150
QUANTITY DIFFERENCE:50
OLD QUANTITY:80
NEW QUANTITY:130
QUANTITY DIFFERENCE:50
OLD QUANTITY:70
NEW QUANTITY:120
QUANTITY DIFFERENCE:50
OLD QUANTITY:130
NEW QUANTITY:180
QUANTITY DIFFERENCE:50
OLD QUANTITY:60
NEW QUANTITY:110
QUANTITY DIFFERENCE:50
OLD QUANTITY:200
NEW QUANTITY:250
QUANTITY DIFFERENCE:50
OLD QUANTITY:200
NEW QUANTITY:250
QUANTITY DIFFERENCE:50
10Quantity Updated
```

CONCLUSION

THANKYOU

National Institute of Technology, Warangal

Project done by -

Sai Kiran Sajja
21CSB0A49

Greata Chakravarty
21CSB0B20