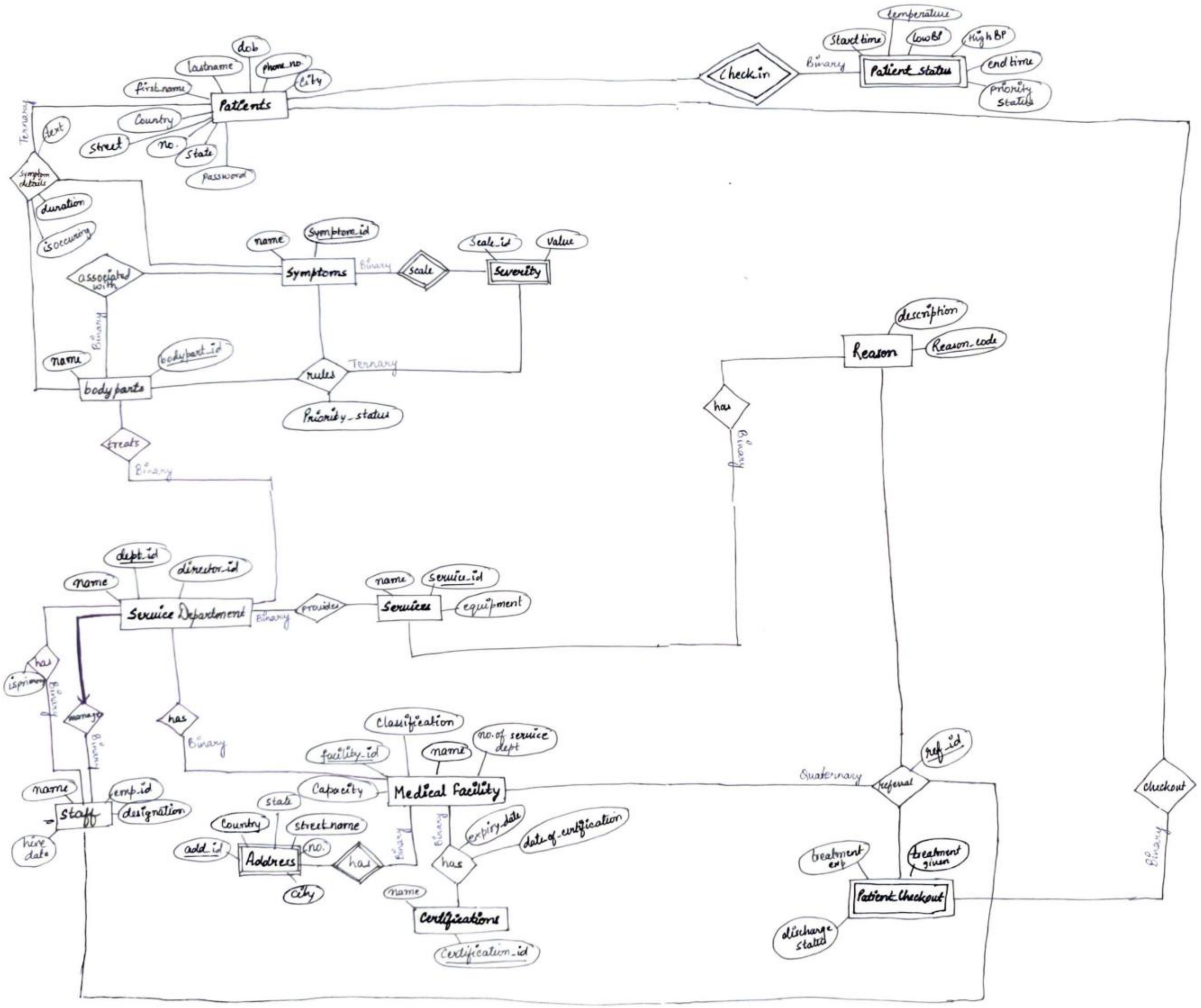
Milestone 1 - Report, due October 20th

For this milestone, you should:

1. Fill in the form for deciding team members as soon as possible.
2. An ER-Diagram along with a list of Entity and Relationship Types that you identify in the project description. For each relationship type, you should state the arity of the relationship e.g. if it is binary, ternary, etc. Relationships should include any hierarchical relationships (subtypes) that you identify. There is no need for verbose text, just a categorized listing of these is fine.
3. Relational Model: A list of tables, 2 - 3 sentences description of each including what constraints (including referential constraints) it encodes, a listing of functional dependencies, a discussion of normal form choices faced and justification for the decision made.
4. For this report, you should list any application constraints (key, participation, other domain constraints) that you identify in the description. A statement acknowledging that you have asked all questions you need to clarify any ambiguities in the description.

Submission by: Bharat Bhardwaj (bbhardw), Garima Garima (ggarima), Himani Jangam (hjangam), Nimisha Tripathi(ntripat)



Tables:

1. **Medical_Facility Table:**

    a.) **Description:** This table stores the details of medical facilities such as name, address, classification of the medical facility, number of service departments, capacity in terms of number of beds.

    b.) **Attributes:**
    1. Facility_id
    2. Classification
    3. Name
    4. Number of Service Departments
    5. Capacity

    c.) **Primary Key:** Facility_id

    d.) **Domain Constraints:**
    1. Facility_id has to be an unsigned integer only and should not be null
    2. Classification can take only codes 1, 2 or 3 which indicate primary, secondary and tertiary respectively and cannot be null
    3. Name has varchar as its datatype and should not be null
    4. Number of Service Departments has to be an unsigned integer only and should not be null
    5. Capacity has to be an unsigned integer only and should not be null

    e.) **Functional Dependencies:** Facility_id → Classification, Name, Number of Service Departments, Capacity

    f.) **Normal Form:** The table will be in BCNF as the primary key Facility_id determines all the other attributes of the table.

2. **Address Table :**

    a.) **Description:** Address table is a weak entity of Medical_Facility Table. It stores the addresses of the all the medical facilitites.

    b.) **Attributes:**
    1. Add_id (Autogenerated id)
    2. Facility_id
    3. Street_name
    4. Street_number
    5. City
    6. State
    7. Country

    c.) **Primary Key:** Add_id, Facility_id

    d.) **Foreign Key:** Facility_id references Medical_Eacility(Facility_id) . On updation and deletion of the facility_id in Medical_Facility table, the facility_id in Address table has to be updated and deleted respectively.

    e.) **Domain Constraints:**
    1. Add_id has to be an unsigned integer only and should not be null

2. Facility_id has to be an unsigned integer only and should not be null
3. Street_name has varchar as its datatype
4. Street_number has to be an unsigned integer only
5. City has varchar as its datatype
6. State has varchar as its datatype
7. Country has varchar as its datatype

f.) Functional Dependencies: Add_id, Facility_id →Street_name, Street_number, City, State, Country

g.) Normal Form: The table will be in BCNF as the primary keys Add_id, Facility_id determine all the other attributes of the table.

3. **Certification Table**:

a.) Description: This table stores the certifications of all the medical facilities

b.) Atributes:

1. Certification_id
2. Name

c.) Primary Key: Certification_id

d.) Domain Constraints:

1. Certification_id has to be an alphanumeric code only, so has varchar datatype and should not be null
2. Name has to be a string and is of varchar datatype

e.) Functional Dependencies: Certification_id → Name

f.) Normal Form: The table will be in BCNF as the primary key Certification_id determines all the other attributes of the table.

4. **Medical_Facility_Certifications Table** :

a.) Description: This table is a binary relationship between Medical_facility table and Certifications table. It stores the certifications done, date of certification and certification expiry date of the medical facilities

b.) Attributes:

1. Certification_id
2. Facility_id
3. Date_of_Certification
4. Expiry_date

c.) Primary Key: Certification_id, Facility_id

d.) Foreign Key: Facility_id references Medical_Eacility(Facility_id) . On updation and deletion of the facility_id in Medical_Facility table, the facility_id in Address table has to be updated and deleted respectively.

e.) Domain Constraints:

1. Certification_id has to be an alphanumeric code only, so has varchar datatype and should not be null
2. Facility_id has to be an unsigned integer only and cannot null
3. Date_of_Certification has to be a date

4. Expiry_date has to be a date

f.) **Functional Dependencies:** Certification_id, Faciltiy_id → Date_of_Certification, Expiry_date

g.) **Normal Form:** The table will be in BCNF as the primary keys Certification_id, Facility_id determine all the other attributes of the table.

5. **Staff_Table :**

a.) **Description:** This table stores the employee details such as name of the employee, designation and date hired.

b.) **Attributes:**

1. Emp_id
2. Name
3. Designation
4. Hire_date

c.) **Primary Key:** Emp_id

d.) **Domain Constraints:**

1. Emp_id has to be an unsigned integer and should not be null.
2. Name has to be a string and takes varchar datatype
3. Designation can take Medical or Non-Medical values only and should not be null
4. Hire_date has to be a date and should not be null

e.) **Functional Dependencies:** Emp_id → name, Designation, Hire_date

f.) **Normal Form:** The table will be in BCNF as the primary key Emp_id determines all the other attributes of the table.

6. **Service_Departments:**

a.) **Description:** It stores the different departments of the medical facilities and the corresponding directors of those departments.

b.) **Attributes:**

1. Dept_id
2. Name
3. Director_id

c.) **Primary Key:** Dept_id

d.) **Foreign Key:** Director_id references Staff(Emp_id). On updation of the Emp_id in Staff table, the Director_id in Address table has to be updated.

e.) **Domain Constraints:**

1. Dept_id has to be an alphanumeric code only, so has varchar datatype and should not be null.
2. Name has to be a string and takes varchar datatype and cannot be null
3. Director_id has to be an unsigned integer and should not be null or default Director_id will be added.

f.) **Functional Dependencies:** Dept_id → Name, Director_id g.) **Normal Form:** The table will be in BCNF as the primary key Dept_id determines all the other attributes of the table.

7. Staff_Service_Departments:

a.) **Description:** This table is a binary relationship between Staff table and Service_Departments table. It stores the service departments in which the staff work. Is_primary is a Boolean attribute that shows if the department in which the employee works is primary (Is_primary =1) or secondary (Is_primary = 0).

b.) **Attributes:**

1. Emp_id
2. Dept_id
3. Is_Primary

c.) **Primary Key:** Emp_id, Dept_id

d.) **Foreign Key:**

1. Emp_id references Staff(Emp_id)
2. Dept_id references Service_Departments(Dept_id)
3. On updation and deletion of the Emp_id in Staff table and Dept_id in Staff_Service_Departments table, Emp_id and Dept_id has to be updated and deleted respectively in Staff_Service_Departments table.

e.) **Domain Constraints:**

1. Emp_id has to be an unsigned integer and should not be null.
2. Dept_id has to be an alphanumeric code only, so has varchar datatype and should not be null.
3. Is_Primary takes boolean (0 or 1) values and is not null. It takes bit datatype

f.) **Functional Dependencies:** Emp_id, Dept_id → Is_Primary

h.) **Normal Form:** The table will be in BCNF as the primary keys Emp_id, Dept_id determine all the other attributes of the table.

8. Medical_Facility_Service_Departments:

a.) **Description:** This table is a binary relationship between Medical_Facility table and Service_Departments table. It stores the service departments present in the medical facilities

b.) **Attributes:**

1. Facility_id
2. Dept_id

c.) **Primary Key:** Faciltiy_id, Dept_id

d.) **Foreign Key:**

1. Dept_id references Service_Departments(Dept_id)
2. Facility_id references Medical_Eacility(Facility_id)
3. On updation and deletion of the Facility_id in Medical_Facility table and Dept_id in Staff_Service_Departments table, Facility_id and Dept_id has to be updated and deleted respectively in Medical_Facility_Service_Departments table.

e.) **Domain Constraints:**

1. Facility_id has to be an unsigned integer only and should not be null
2. Dept_id has to be an alphanumeric code only, so has varchar datatype and should not be null.

f.) Functional Dependencies: No dependencies

g.) Normal Form: The table will be in BCNF as there are no functional dependencies

9. Services Table :

a.) Description: This table stores the different services provided in medical facilities.

b.) Attributes:

1. Service_id
2. Name
3. Equipments

c.) Primary Key: Service_id

d.) Domain Constarints:

1. Service_id has to be an alphanumeric code only, so has varchar datatype and should not be null
2. Name has to be a string and takes varchar datatype and cannot be null
3. Equipmet has to be a string and is of varchar datatype

e.) Functional Dependencies: Service_id →Name, Equipments

f.) Normal Form: The table will be in BCNF as the primary key Service_id determines all the other attributes of the table.

10. Service_Departments_Services:

a.) Description: This table is a binary relationship between Services table and Service_Departments table. It stores the different services provided by the service departments present in the medical facilities.

b.) Attributes:

1. Dept_id
2. Service_id

c.) Primary Key: Dept_id, Service_id

d.) Foreign Key:

1. Dept_id references Services_Departments(Dept_id)
2. Service_id references Services(Service_id)
3. On updation and deletion of the Dept_id in Services_Departments table and Service_id in Services table, Dept_id and Service_id has to be updated and deleted respectively in Service_Departments table.

e.) Domain Constraints:

1. Service_id has to be an alphanumeric code only, so has varchar datatype and should not be null
2. Dept_id has to be an alphanumeric code only, so has varchar datatype and should not be null

f.) Functional Dependencies: No dependencies g.) **Normal Form:** The table will be in BCNF as there are no functional dependencies

11. **Body_Parts Table:**

a.) **Description:** This table stores the body part name and its id

b.) **Attributes:**
1. Part_id
2. Name

c.) **Primary Key:** Part_id

d.) **Domain Constraints:**
1. Part_id has to be an alphanumeric code, it takes varchar as datatype and can not be null
2. Name has to be a string and takes varchar datatype and cannot be null

e.) **Functional Dependencies:** Part_id →Name

g.) **Normal Form:** The table will be in BCNF as the primary key Part_id determines all the other attributes of the table.

12. **Body_Parts_Service_Departments Table:**

a.) **Description:** This table is a binary relationship between Body_Parts table and Service_Departments table. It stores the different body parts that the service departments provide treatment in.

b.) **Attributes:**
1. Part_id
2. Dept_id

c.) **Primary Key:** Part_id, Dept_id

d.) **Foreign Key:**
1. Part_id references Body_Parts(part_id)
2. Dept_id references Services_Departments(Dept_id)
3. On updation and deletion of the Dept_id in Services_Departments table and Part_id in Body_Parts table, Dept_id and Parts_id has to be updated and deleted respectively in Body_Parts_Service_Departments table.

e.) **Domain Constraints:**
1. Part_id has to be an alphanumeric code, it takes varchar as datatype and cannot be null
2. Dept_id has to be an alphanumeric code only, so has varchar datatype and should not be null

f.) **Functional Dependencies:** No dependencies

g.) **Normal Form:** The table will be in BCNF as there are no functional dependencies

13. **Symptoms Table:**

a.) **Description:** This table stores the different symptoms name and their code

b.) **Attributes:**

1. Sym_id
2. Name

c.) Primary Key: Sym_id

d.) Domain Constraints:

1. Sym_id has to be an alphanumeric code, it takes varchar as datatype and cannot be null
2. Name has to be a string and takes varchar datatype and cannot be null

e.) Functional Dependencies: Sym_id → Name

f.) Normal Form: The table will be in BCNF as the primary key Sym_id determines all the other attributes of the table.

14. Symptoms_Body_Parts Table:

a.) Description: This table is a binary relationship between Symptoms table and Body_Parts table. This table stores the body parts associated with the symptoms.

b.) Attributes:

1. Sym_id
2. Part_id

c.) Primary Key: Sym_id, Part_id

d.) Foreign Key:

1. Sym_id references Symptoms(Sym_id)
2. Part_id references Body_Parts(Part_id)
3. On updation and deletion of the Sym_id in Symptoms table and Part_id in Body_Parts table, Sym_id and Part_id has to be updated and deleted respectively in Symptoms_Body_Parts table.

e.) Domain Constraints:

1. Sym_id has to be an alphanumeric code, it takes varchar as datatype and cannot be null
2. Part_id has to be an alphanumeric code, it takes varchar as datatype and cannot be null

f.) Functional Dependencies: No dependencies

g.) Normal Form: The table will be in BCNF as there are no functional dependencies

15. Severity Table:

a.) Description: This table stores all the possible static severity values like range of severity from 1 to 10 and heavy, Normal or Light severity values

b.) Attributes:

1. Svr_id
2. Value

c.) Primary Key: Svr_id

d.) Domain Constraints:

1. Svr_id has to be an integer only and should not be null
2. Value can take string input and has varcjar datatype

e.) Functional Dependencies: Svr_id → Value

f.) Normal Form: The table will be in BCNF as the primary key Svr_id determines all the other attributes of the table.

16. **Scale Table:**

a.) Description: This table stores the different severity scales associated with the symptoms.

b.) Attributes:

1. Sym_id
2. Svr_id

c.) Primary Key: Sym_id. Svr_id

d.) Foreign Key:

1. Sym_id references Symptoms(Sym_id)
2. Svr_id references Severity(Svr_id)
3. On updation and deletion of the Sym_id in Symptoms table and Svr_id in Severity table, Sym_id and Svr_id has to be updated and deleted respectively in Scale table.

e.) Domain Constraints:

1. Sym_id has to be an alphanumeric code, it takes varchar as datatype and cannot be null
2. Svr_id has to be an integer only and should not be null

f.) Functional Dependencies: No dependencies

h.) Normal Form: The table will be in BCNF as there are no functional dependencies

17. **Rules Table:**

a.) Description: Rules table is a ternary relationship of Body_Parts table, Symptoms table and Severity table. This table stores the priority status based on the body part symptom and severity values. New rules can be inserted into this table, whenever required.

b.) Attributes:

1. Part_id
2. Sym_id
3. Svr_id
4. Priority Status

c.) Primary Key: Part_id, Sym_id, Svr_id

d.) Foreign Key:

1. Sym_id references Symptoms(Sym_id)

2. Part_id references Body_Parts(part_id)
3. Svr_id references Severity(Svr_id)
4. On updation and deletion of the Sym_id in Symptoms table or Svr_id in Severity table or Part_id in Body_Parts table, Sym_id, Svr_id and Part_id has to be updated and deleted respectively in Rules table.

e.) Domain Constraints:

1. Sym_id has to be an alphanumeric code, it takes varchar as datatype and cannot be null
2. Svr_id has to be an integer only and should not be null
3. Part_id has to be an alphanumeric code, it takes varchar as datatype and cannot be null
4. Priority_Status can take value High, Normal, Quarantine only

f.) Functional Dependencies: Part_id, Sym_id, Svr_id → Priority_Status

g.) Normal Form: The table will be in BCNF as the primary key determines all the other attributes of the table.

18. Patients Table:

a.) Description: It stores the registered patient details

b.) Attributes:

1. Patient_id
2. First_name
3. Last_name
4. DOB
5. Phone_no
6. House_number
7. Stree_name
8. City
9. State
10. Country

c.) Primary Key: Patient_id, Last_name, DOB, City

d.) Domain Constraints:

1. Patient_id takes integer values only and cannot be null. It is auto generated in the table.
2. First_name, Last_name have to be strings and are of varchar datatype and cannot be null
3. DOB has to be a date and cannot be null
4. Phone_no. has to be string only and takes varchar(15) datatype
5. House_number, Street_name, City, State and Country have to be strings and take varchar datatype.

e.) Functional Dependencies: Patient_id, Last_name, DOB,City → First_name, Phone_no, House_number, Street_name, City, State and Country.

h.) Normal Form: The table will be in BCNF as the primary key determines all the other attributes of the table.

19. Symptom_details:

a.) Description: This table is a ternary relationship of Body_Parts table, Symptoms table and Patients table. This table stores the symptom details of the patients that come for treatment.

b.) Attributes:

1. Part_id
2. Sym_id
3. Patient_id
4. Is_occuring
5. Duration
6. Text

c.) Primary Key: Part_id, Sym_id, Patient_id

d.) Foreign Key:

1. Patient_id references Patients(Patient_id)
2. Sym_id references Symptoms(Sym_id)
3. Part_id references Body_Parts(part_id)
4. On updation and deletion of the Sym_id in Symptoms table or Part_id in Body_Parts table and Patient_id in Patients table, Sym_id, Part_id and Patient_id has to be updated and deleted respectively in Symptom_details table.

e.) Domain Constraints:

1. Sym_id has to be an alphanumeric code, it takes varchar as datatype and cannot be null
2. Part_id has to be an alphanumeric code, it takes varchar as datatype and cannot be null
3. Patient_id takes integer values only and cannot be null.
4. Is_occuring is a boolean attribute which can take only 0 or 1.
5. Duration has to be an unsigned integer only and should not be null
6. Text is a string and takes text datatype

f.) Functional Dependencies: Part_id, Sym_id, Patient_id → Is_occuring, Duration, Text

i.) Normal Form: The table will be in BCNF as the primary key determines all the other attributes of the table.

20. Patient_Status Table:

a.) Description: This table is a weak entity of Patients table and stores the patient check-in details

b.) Attributes:

1. Patient_id
2. Start_time
3. End_time
4. LowBP
5. HighBP
6. Temperature
7. Priority_status

c.) Primary Key: Patient_id

d.) Foreign Key:

1. Patient_id references Patients(Patient_id)
2. On updation and deletion of Patient_id in Patients table, Patient_id has to be updated and deleted respectively in Patient_Status table.

e.) Domain Constraints:

1. Patient_id takes integer values only and cannot be null.
2. Start_time and End_time take DateTime values only
3. LowBP and HighBP values are of unsigned Tinyint datatype
4. Temperature takes Decimal(3,2) datatype
5. Priority_Status can take value High, Normal, Quarantine only

f.) Functional Dependencies: Patient_id → Start_time, End_time, LowBP, HighBP, Temperature, Priority_Status

g.) Normal Form: The table will be in BCNF as the primary key determines all the other attributes of the table.

21. Reason Table:

a.) Description: Table to store the reasons for referring the patient to some other medical facility

b.) Attributes:

1. Reason_code
2. Description

c.) Primary Key: Reason_code

d.) Domain Constraints:

1. Reason_code can take only 1, 2 or 3 values
2. Description has fixed values Service unavailable at time of visit, Service not present at facility and Non payment.

e.) Functional Dependencies: Reason_code → Description

h.) Normal Form: The table will be in BCNF as the primary key determines all the other attributes of the table.

22. Patient_Checkout:

a.) Description: This table stores the patient checkout details and it is a weak entity of Patients table

b.) Attributes:

1. Patient_id
2. Discharge_Status
3. Treatment_given
4. Experience
5. Referral_id

c.) Primary Key: Patient_id

d.) Foreigh Key:

1. Referral_id reference Referral(Referral_id)
2. Patient_id references Patients(Patient_id)
3. On updation and deletion of Patient_id in Patients table, Referral_id in Referral table, Patient_id, Referral_id has to be updated and deleted respectively in Patient_Checkout table.

e.) Domain Constraints:

1. Patient_id takes integer values only and cannot be null.
2. Discharge_Status can take only values Treated Successfully, Deceased and Referred
3. Treatment_given is a text datatype
4. Experience is a text datatype and takes values Misdiagnosis, Patient acquired infection during hospital stay
5. Referral_id takes unsigned integer values

f.) Functional Dependencies: Patient_id → Discharge_Status, Treatment_given, Experience, Referral_id

g.) Normal Form: The table will be in BCNF as the primary key determines all the other attributes of the table

23. Referral:

a.) Description: This table is a terenary relationship of Patients table, Staff table, Medical_facility table and stores the patient ids who are referred to some other medical facility by employees.

b.) Attributes:

1. Referral_id
2. Patient_id
3. Emp_id
4. Facility_id

c.) Primary Key: Referral_id

d.) Foreign Key:

1. Emp_id references Staff(Emp_id)
2. Patient_id references Patients(Patient_id)
3. Facility_id references Medical_Eacility(Facility_id)
4. On updation and deletion of Patient_id in Patients table, Emp_id in Staff table, Facility_id in Medical_Facility table, Patient_id, Emp_id, Facility_id has to be updated and deleted respectively in Referral table.

e.) Domain Constraints:

1. Patient_id takes integer values only and cannot be null.
2. Facility_id has to be an unsigned integer only and should not be null
3. Emp_id has to be an unsigned integer and should not be null.
4. Referral_id takes unsigned integer values

f.) Functional Dependencies: Referral_id → Patient_id, Emp_id, Facility_id

g.) Normal Form: The table will be in BCNF as the primary key determines all the other attributes of the table