LAU
School of Arts and Sciences

Department of Computer Science & Mathematics

THE CLINIC OF LOST SOULS

by

BYTE BUILDERS

DANA ITANI
dana.itani05@lau.edu

NICOLAS TALJ
nicolas.talj@lau.edu

NUR EL HUDA EL UMARI
nurelhuda.elumari@lau.edu

TALA HACHEM
tala.hachem01@lau.edu

TALA SABRA
tala.sabra@lau.edu

A REPORT

submitted to Dr. Ramzi R. Haraty in partial fulfillment of the requirements for the course "CSC375: Database Management Systems" in Computer Science

Phase III – 18503 words – 148 pages

December 09, 2024

This page is intentionally left blank.

I. Table of Contents:

I. TABLE OF CONTENTS: 3

II. PHASE IV REQUIREMENTS: 12

III. INTRODUCTION: 13

IV. COPYRIGHT NOTICE: 14

V. SYSTEM DESCRIPTION AND CONSTRAINTS: 15

VI. ER DIAGRAM FOR THE CLINIC OF LOST SOULS: 16

VII. ENTITIES: 17

1. DOCTOR 17
2. NURSE 18
3. PATIENT 19
4. DEPARTMENT 20
5. APPOINTMENT 21
6. TREATMENT 22
7. DIAGNOSIS 23
8. MEDICAL_TEST 24
9. PRESCRIPTION 25
10. PHARMACY 26
11. ALLERGY 27
12. MEDICAL_HISTORY 28
13. MEDICATION 29
14. ROOM 30
15. MEDICAL_EQUIPMENT 31
16. BILLING 32

VIII.RELATIONSHIPS: 33

1. BELONGS_TO1 33
2. BELONGS_TO2 33
3. SERVICES 34
4. VISITS 34

5. REQUIRES 35
6. HAS A1 36
7. HAS A2 36
8. HAS A3 36
9. HAS A4 37
10. ISSUES 37
11. CONTAINS1 38
12. CONTAINS2 38
13. SUPPLIES 39
14. PERFORMS 39
15. TREATS 40
16. OCCURS_IN 40
17. UNDERGOES 41
18. RESIDES_IN 41
19. PAYS 42
IX. ER TO RELATIONAL MAPPING ALGORITHM: 43
STEP 1: Mapping of Regular Entity Types 44
1. DOCTOR 44
2. NURSE 44
3. PATIENT 45
4. DEPARTMENT 45
5. APPOINTMENT 45
6. TREATMENT 46
7. DIAGNOSIS 46
8. MEDICAL_TEST 46
9. PHARMACY 47
10. ALLERGY 47
11. MEDICAL_HISTORY 47
12. MEDICATION 48
13. ROOM 48
14. MEDICAL_EQUIPMENT 48
15. BILLING 49
STEP 2: Mapping of Weak Entity Types 50
1. PRESCRIPTION 50

STEP 3: Mapping of Binary 1:1 Relationship Types 51

1. MEDICAL_HISTORY (HAS A1) 51

2. BILLING (PAYS) 51

STEP 4: Mapping of Binary 1:N Relationship Types 52

1. DOCTOR (BELONGS_TO1) 52

2. ALLERGY (HAS_A2) 52

3. APPOINTMENT (HAS_A3) 52

4. DIAGNOSIS (HAS_A4) 53

5. PRESCRIPTION (ISSUES) 53

6. MEDICAL_EQUIPMENT (CONTAINS2) 53

7. MEDICATION (SUPPLIES) 53

8. TREATMENT (PERFORMS) 54

9. TREATMENT (OCCURS_IN) 54

10. PATIENT (RESIDES_IN) 54

STEP 5: Mapping of Binary M:N Relationship Types 55

1. BELONGS_TO2 55

2. SERVICES 55

3. VISITS 55

4. REQUIRES 56

5. CONTAINS1 56

6. TREATS 56

7. UNDERGOES 57

STEP 6: Mapping of Multivalued Attributes 58

1. DOCTOR_PHONE_NUMBER 58

2. NURSE_PHONE_NUMBER 58

3. PATIENT_PHONE_NUMBER 58

STEP 7: Mapping of N-ary Relationship Types 59

Final Step: Final Displays 60

DOCTOR 60

NURSE 60

PATIENT 60

DEPARTMENT 60

APPOINTMENT 60

TREATMENT 60

DIAGNOSIS 60

MEDICAL_TEST 60
PHARMACY 60
ALLERGY 61
MEDICAL_HISTORY 61
MEDICATION 61
ROOM 61
MEDICAL_EQUIPMENT 61
BILLING 61
PRESCRIPTION 61
MEDICAL_HISTORY (HAS_A1) 61
BILLING (PAYS) 62
DOCTOR (BELONGS_TO1) 62
ALLERGY (HAS_A2) 62
APPOINTMENT (HAS_A3) 62
DIAGNOSIS (HAS_A4) 62
PRESCRIPTION (ISSUES) 62
MEDICAL_EQUIPMENT (CONTAINS2) 62
MEDICATION (SUPPLIES) 62
TREATMENT (PERFORMS) 62
TREATMENT (OCCURS_IN) 63
PATIENT (RESIDES_IN) 63
BELONGS_TO2 63
SERVICES 63
VISITS 63
REQUIRES 63
CONTAINS1 63
TREATS 63
UNDERGOES 63
DOCTOR_PHONE_NUMBER 64
NURSE_PHONE_NUMBER 64
PATIENT_PHONE_NUMBER 64
X. TABLE STRUCTURE FOR THE CLINIC OF LOST SOULS: 65
1. Doctor 65
2. Nurse 66

3. Patient 66

4. Department 67

5. Appointment 67

6. Treatment 67

7. Diagnosis 68

8. Medical_Test 68

9. Pharmacy 69

10. Allergy 69

11. Medical_History 70

12. Medication 70

13. Room 71

14. Medical_Equipment 71

15. Billing 72

16. Prescription 72

17. Belongs_To2 73

18. Services 73

19. Visits 74

20. Requires 74

21. Contains1 74

22. Treats 75

23. Undergoes 75

24. Doctor_Phone_Number 75

25. Nurse_Phone_Number 76

26. Patient_Phone_Number 76

XI. TABLE DESCRIPTIONS: 78

1. Doctor 79

2. Nurse 80

3. Patient 81

4. Department 81

5. Appointment 82

6. Treatment 82

7. Diagnosis 83

8. Medical_Test 83

9. Pharmacy 84

10. Allergy 84
11. Medical_History 85
12. Medication 85
13. Room 86
14. Medical_Equipment 86
15. Billing 87
16. Prescription 87
17. Belongs_To2 88
18. Services 88
19. Visits 88
20. Requires 89
21. Contains1 89
22. Treats 89
23. Undergoes 90
24. Doctor_Phone_Number 90
25. Nurse_Phone_Number 90
26. Patient_Phone_Number 91
XII. INSERTING DATA: 92
1. Doctor 92
2. Nurse 93
3. Patient 94
4. Department 95
5. Appointment 95
6. Treatment 96
7. Diagnosis 97
8. Medical_Test 98
9. Pharmacy 99
10. Allergy 100
11. Medical_History 101
12. Medication 102
13. Room 103
14. Medical_Equipment 104
15. Billing 105
16. Prescription 106

17. Belongs_To2 107
18. Services 108
19. Visits 109
20. Requires 109
21. Contains1 110
22. Treats 110
23. Undergoes 111
24. Doctor_Phone_Number 112
25. Nurse_Phone_Number 112
26. Patient_Phone_Number 113
XIII.FINAL TABLES STATE: 114
1. Doctor 114
2. Nurse 114
3. Patient 114
4. Department 115
5. Appointment 115
6. Treatment 116
7. Diagnosis 116
8. Medical_Test 116
9. Pharmacy 117
10. Allergy 117
11. Medical_History 117
12. Medication 118
13. Room 118
14. Medical_Equipment 119
15. Billing 119
16. Prescription 120
17. Belongs_To2 120
18. Services 121
19. Visits 121
20. Requires 122
21. Contains1 122
22. Treats 123
23. Undergoes 123

24. Doctor_Phone_Number 124
25. Nurse_Phone_Number 124
26. Patient_Phone_Number 125

XIV. QUERIES: 126

Query 1: The Case of the Hopeless Romantic Patient 126
Query 2: The Case of the Missing Wallet 126
Query 3: The Case of the Vengeful Doctor 127
Query 4: The Case of the Dedicated Doctors 128
Query 5: The Case of Duplicate Identification Diagnosis 129
Query 6: Identifying Patients with Birthday Discount or Free Service 130
Query 7: The Emergency Blood Rush 131
Query 8: The Case of Canceling An Appointment 132
Query 9: Prioritizing Patients Requiring Immediate Treatment 133
Query 10: The Case of the Missing Defibrillator 134

XV. NORMALIZATION UP TO THE BCNF NORMAL FORM: 135

1. Doctor 137
2. Nurse 137
3. Patient 138
4. Department 138
5. Appointment 139
6. Treatment 139
7. Diagnosis 140
8. Medical_Test 140
9. Pharmacy 141
10. Allergy 141
11. Medical_History 142
12. Medication 142
13. Room 143
14. Medical_Equipment 143
15. Billing 144
16. Prescription 144
17. Belongs_To2 145
18. Visits 145
19. Undergoes 146

Relation Schemas without non-prime attributes: 146

20. Services 146

21. Requires 146

22. Contains1 146

23. Treats 147

24. Doctor_Phone_Number 147

25. Nurse_Phone_Number 147

26. Patient_Phone_Number 147

XVI. CONCLUSION: 148

II. Phase IV Requirements:

- Normalize your database up to the BCNF Normal Form.
- Hand in the complete project (including all the phases).

III. Introduction:

We, “BYTE BUILDERS,” take pleasure in submitting the first phase of our project entitled “THE CLINIC OF LOST SOULS.” This work project provided us the opportunity to experience the real flavor of creating a database and the craft of designing ER diagrams.

In our report, we summarize what we have done during our work process. We were assigned to create a database design for a clinic. Overall, our work experience has been constructive and useful for our career. We’ve gained experience by working with Microsoft Visio, and many other tools. This report will explore the different aspects of what we have done when designing the database for The Clinic of Lost Souls.

IV. Copyright Notice:

V. System Description and Constraints:

Mission Statement from The Clinic of Lost Souls

Our clinic is not just a normal clinic, but it is also a mysterious sanctuary where patients embark on a journey to reclaim parts of their missing souls and heal. At some point in life, everyone feels like there's a void in their hearts that they can't seem to comprehend. Well, if you ever feel like that, do not be distressed because there's a clinic that just opened, and it specializes in treating the forgotten or missing parts of your soul.

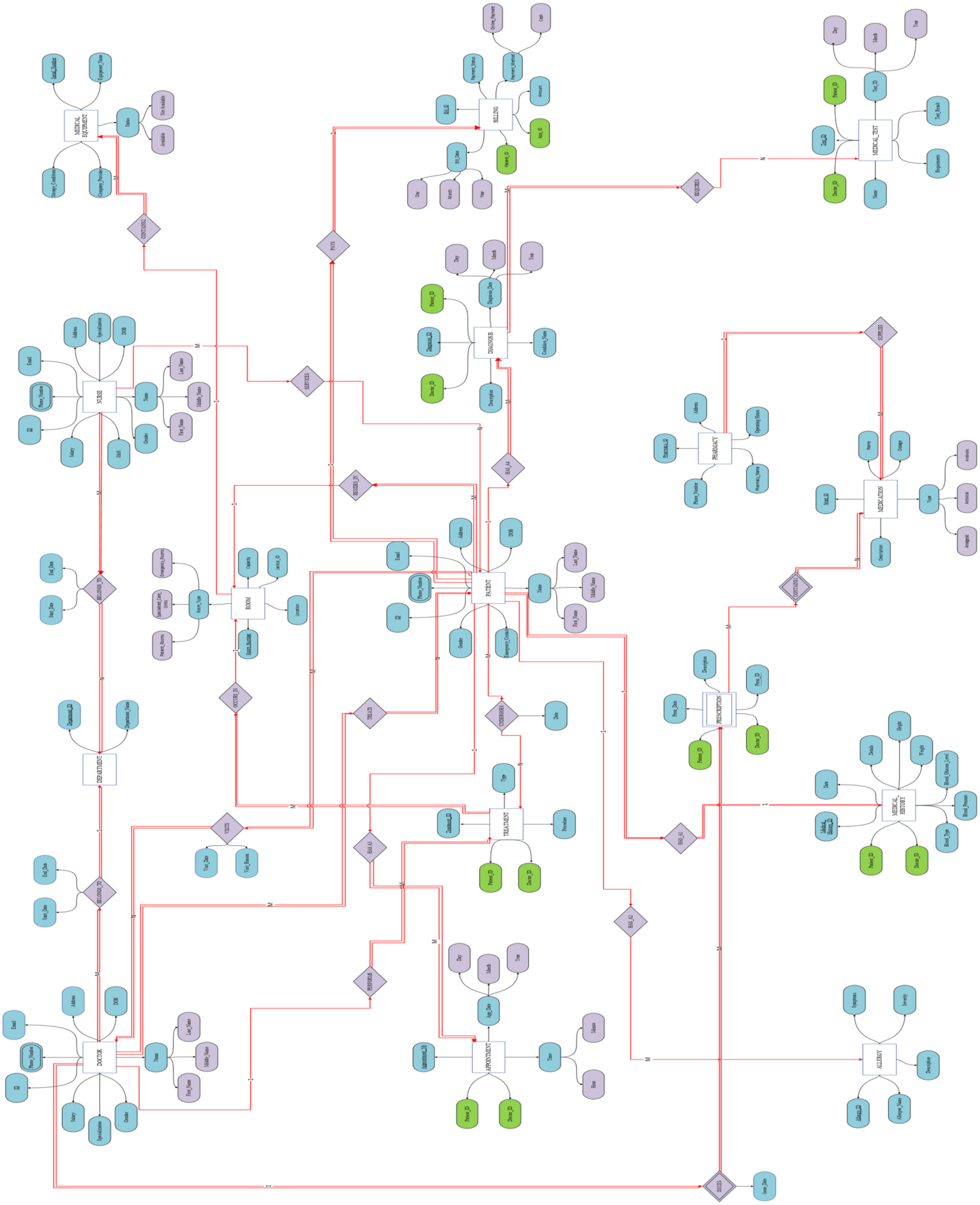
Our clinic provides care for both patients with visible diseases and ones who feel incomplete or broken. Whether you suffer from a neglected passion, a broken heart, a lost sense of purpose, a forgotten dream, overwhelming trauma, fading memories of a loved one, or even chronic illnesses, we have the perfect treatment plan for you. We exist solely for our patients to be seen, valued, and taken care of, regardless of where they come from and the difficulty of their treatment. We are here to make the impossible possible. The Clinic of Lost Souls has specialized departments dedicated to healing our patients physically, mentally, and emotionally.

Our doctors will guide our patients on a journey to rediscover their lost selves and find their worth, whereas our nurses will heal their inner child, mend their broken hearts, and provide the support they need along the way. Our staff are chosen carefully to make our patients' journey as comfortable as possible. We conduct special tests that patients should undergo before diagnosis, such as gazing into the "Pond of Wonders," which reflects our patients' emotions by illuminating the pond with various colors. Our clinic also provides special kinds of therapy, such as nature therapy, where patients have therapy sessions in the garden of the clinic, which is filled with magical flowers and plants. The soothing scents of these plants ease our patients' worries, relax their minds, and warm their souls. Additionally, we provide pet therapy, which allows our patients to spend time with adorable animals -dogs, cats, and singing birds that bring joy and lift their spirits.

You will never feel lonely and unheard in our clinic because we established a supportive community of lost souls where patients share their struggles with other patients who have successfully healed their souls. As a result, you will be granted an opportunity to meet new people and form deep connections with fellow spirit companions throughout your magical journey.

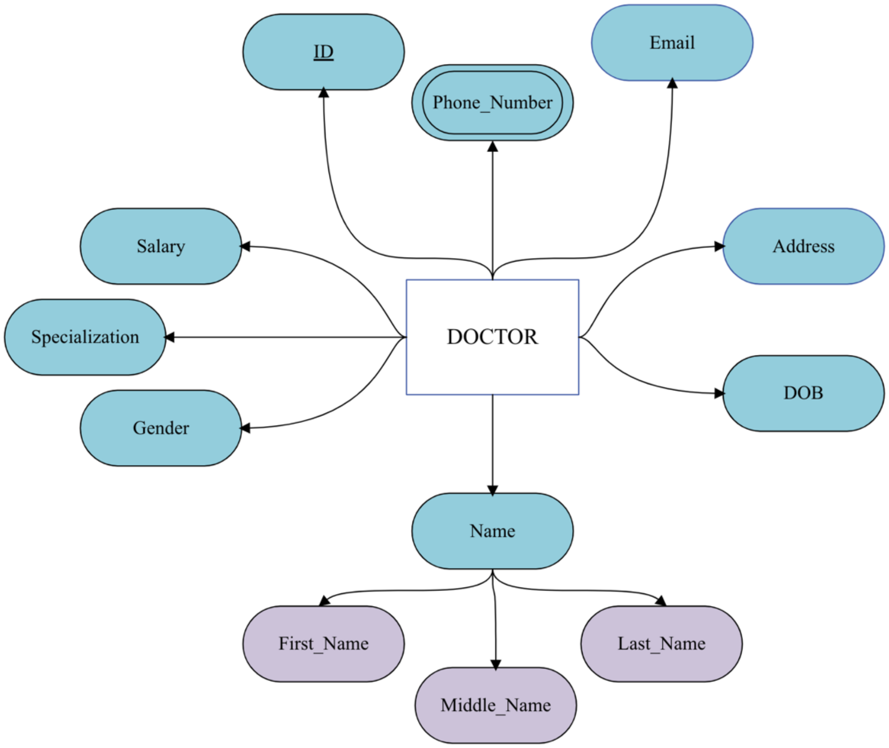
All you need to do is book an appointment and we'll welcome you with open arms. You will have to take medical tests pre- or post-appointment for our doctors to precisely diagnose your medical condition and guide you toward the best treatment. You will be given a prescription written by our experienced doctors specifying the treatment plan, therapies that might help, and medication if necessary. If your treatment requires staying in the clinic, our nurses will cater to all your needs, making your stay as comfortable as possible in our specialized patient room.

VI. ER diagram for The Clinic of Lost Souls:



VII. Entities:

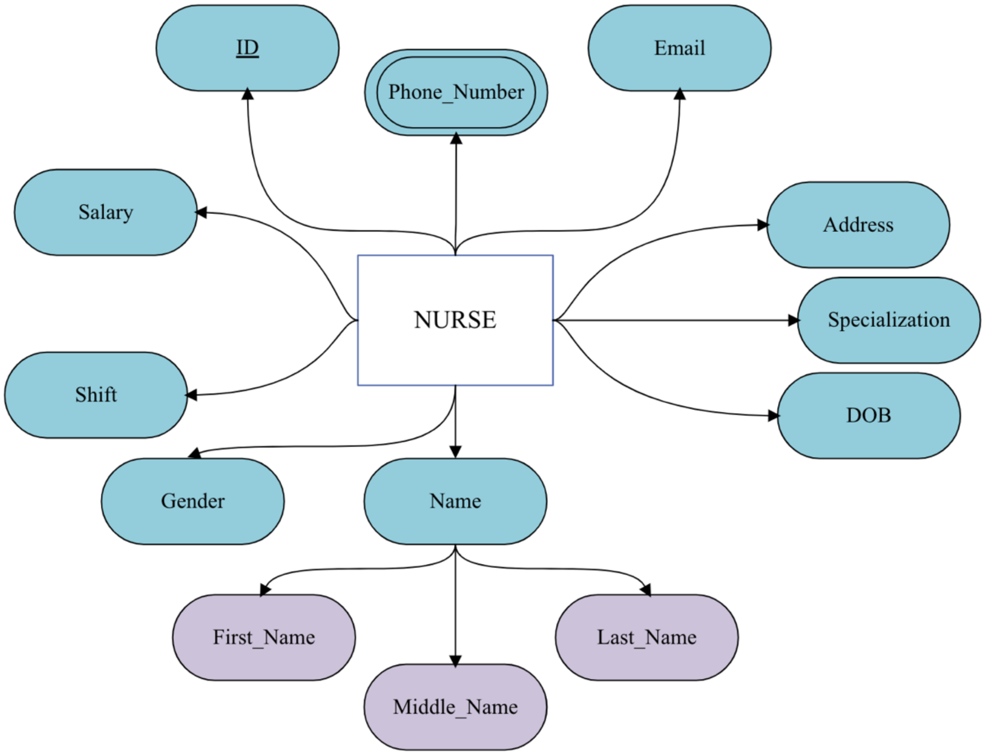
1. DOCTOR



The doctors are the building blocks of our clinic, fundamentally responsible for diagnosing, treating, and managing their patients' health. They work diligently to ensure that each patient's treatment journey is comfortable. Their expertise is essential to our commitment to providing exceptional care. The **DOCTOR**'s primary key is the **ID**, serving as a unique identifier for each doctor.

- Other Attributes:
  - **Name**: A composite attribute that provides the full name of the doctor and is composed of:
    - **First_Name**
    - **Middle_Name**
    - **Last_Name**
  - **Phone_Number**: A multivalued attribute that serves as the contact information for the doctor.
  - **Email**: A formal and professional form of communication used in our clinic.
  - **Address**: Stores the residential address of the doctor.
  - **DOB**: Stores the date of birth of our esteemed doctor in the format: MM-DD-YYYY.
  - **Gender**: The gender of the doctor: male or female.
  - **Specialization**: Stores the field of expertise of each doctor.
  - **Salary**: Stores the salary of the doctor.

2. NURSE

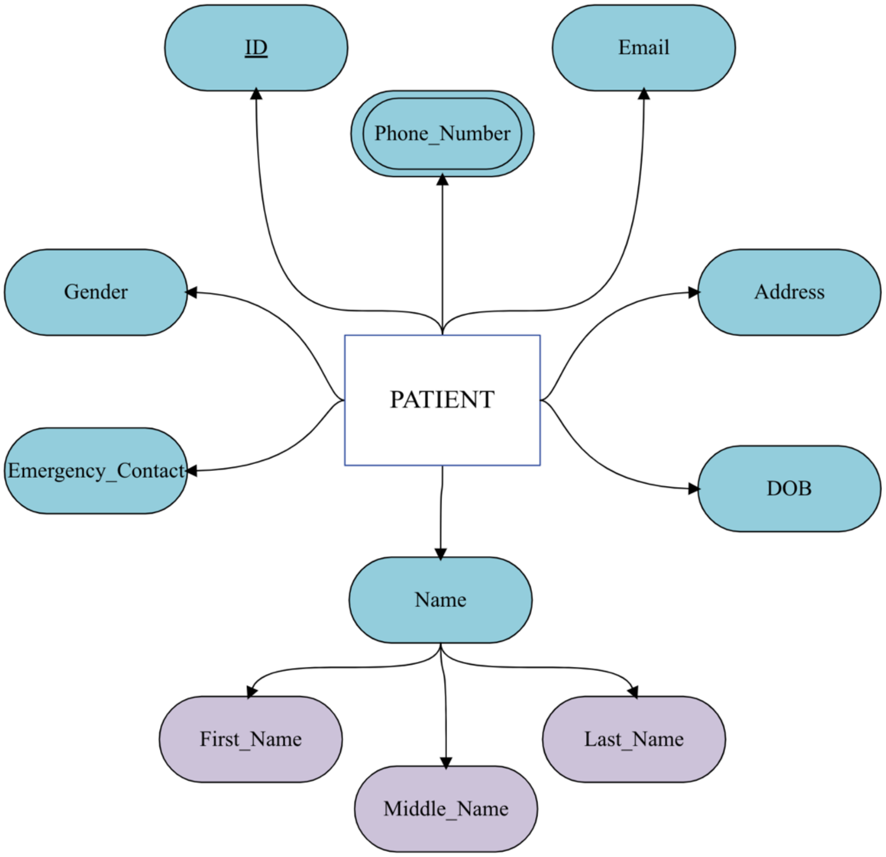


The nurses are the compassionate caregivers in our clinic, and their presence is crucial for maintaining a positive atmosphere. Without them, both patients and doctors would experience significant distress. Nurses play a fundamental role by providing medical care, assistance, and comfort to our patients while helping doctors monitor patient vitals and assisting with treatments. The **NURSE**'s primary key is the **ID**, serving as a unique identifier for each nurse.

- Other Attributes:
  - **Name**: A composite attribute that provides the full name of the nurse and is composed of:
    - **First_Name**
    - **Middle_Name**
    - **Last_Name**
  - **Phone_Number**: A multivalued attribute that serves as the contact information for the nurse.
  - **Email**: A formal and professional form of communication used in our clinic.
  - **Address**: Stores the residential address of the nurse.
  - **Specialization**: Some nurses might be specialized in certain medical fields and are assigned to work in that field.
  - **DOB**: Stores the date of birth of our caring nurse in the format: MM-DD-YYYY.
  - **Gender**: The gender of the nurse: male or female.
  - **Shift**: Stores time when the nurse is available to work in the clinic.

- **Salary**: Stores the salary of the nurse.

3. PATIENT

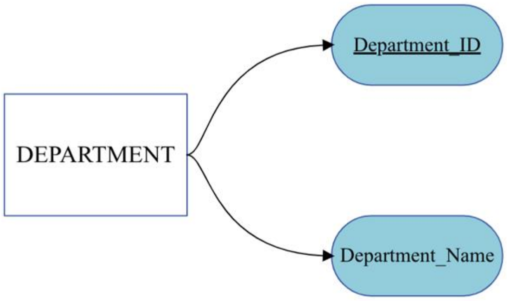


The patients are why our clinic was built, and we take great care in providing the best for all our distinguished patients. Our clinic's mission is to make our patients' treatment a comfortable and joyful journey. The **PATIENT**'s primary key is the **ID,** serving as a unique identifier for each patient.

- Other Attributes:
  - **Name**: A composite attribute that provides the full name of the patient and is composed of:
    - **First_Name**
    - **Middle_Name**
    - **Last_Name**
  - **Phone_Number**: A multivalued attribute that serves as the contact information for the patient.
  - **Email**: A formal and professional form of communication used in our clinic between patients and doctors for any inquiries.
  - **Address**: Stores the residential address of the patient.
  - **DOB**: Stores the date of birth of our distinguished patient in the format: MM-DD-YYYY.

- **Emergency_Contact**: Stores the contact information of the guardian or spouse of the patient.
- **Gender**: The gender of the patient: male or female.

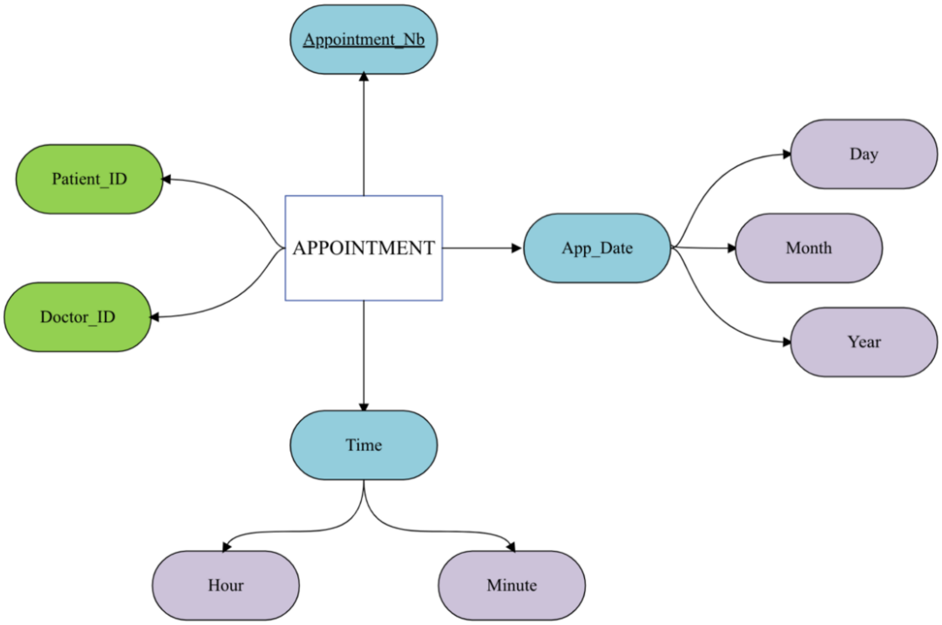
4. DEPARTMENT



The **DEPARTMENT** entity represents distinct divisions within the healthcare facility, such as cardiology, pediatrics, and radiology. It keeps record of the departments in our clinic. The **DEPARTMENT**'s primary key is **Department_ID**, a unique identifier for each department.

- Other Attributes:
  - **Department_Name**: Stores the name of the department (e.g., Cardiology, Pediatrics…).

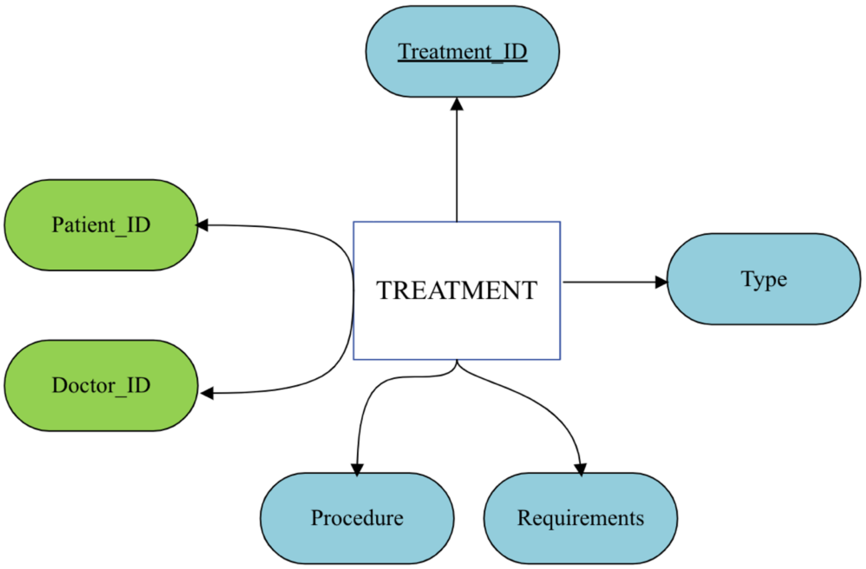
5. APPOINTMENT



An appointment is a scheduled meeting between a patient and a doctor at a clinic. It records essential information about the visit, such as date and time, and associates the visit with a patient and a doctor. The **APPOINTMENT**'s primary key is the **Appointment_Nb**, a unique identifier for each appointment.

- Other Attributes:
  - **App_Date**: The date of the appointment in the format MM/DD/YYYY.
  - **Time**: The time of the appointment, stored in hours HH/MM/SS (hours and minutes) format.
- Foreign Keys:
  - **Patient_ID**: A reference to the patient attending the appointment.
  - **Doctor_ID**: A reference to the doctor conducting the appointment.

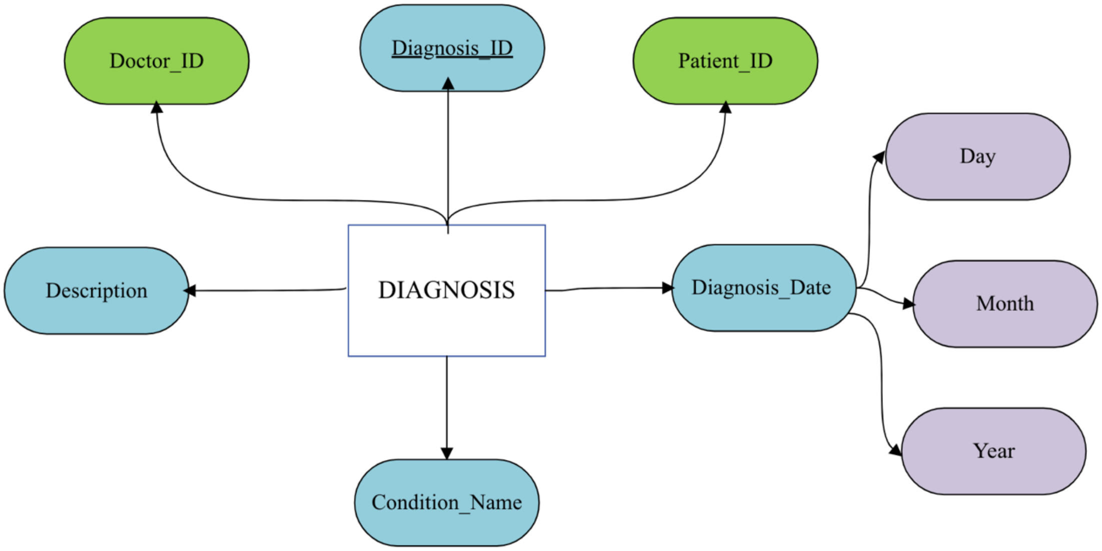
6. TREATMENT



The **TREATMENT** entity stores information on treatments provided to patients, detailing the type of treatment and any specific procedures involved. The **TREATMENT**'s primary key is **Treatment_ID**, a unique identifier for each treatment.

- Other Attributes:
  - **Type**: Specifies the type of treatment, such as medication, surgery, or therapy.
  - **Procedure**: Describes the procedure used in the treatment.
  - **Requirements**: Any prerequisites or required conditions for the treatment.
- Foreign Keys:
  - **Patient_ID**: A reference to the patient receiving the treatment.
  - **Doctor_ID**: A reference to the doctor administering the treatment.

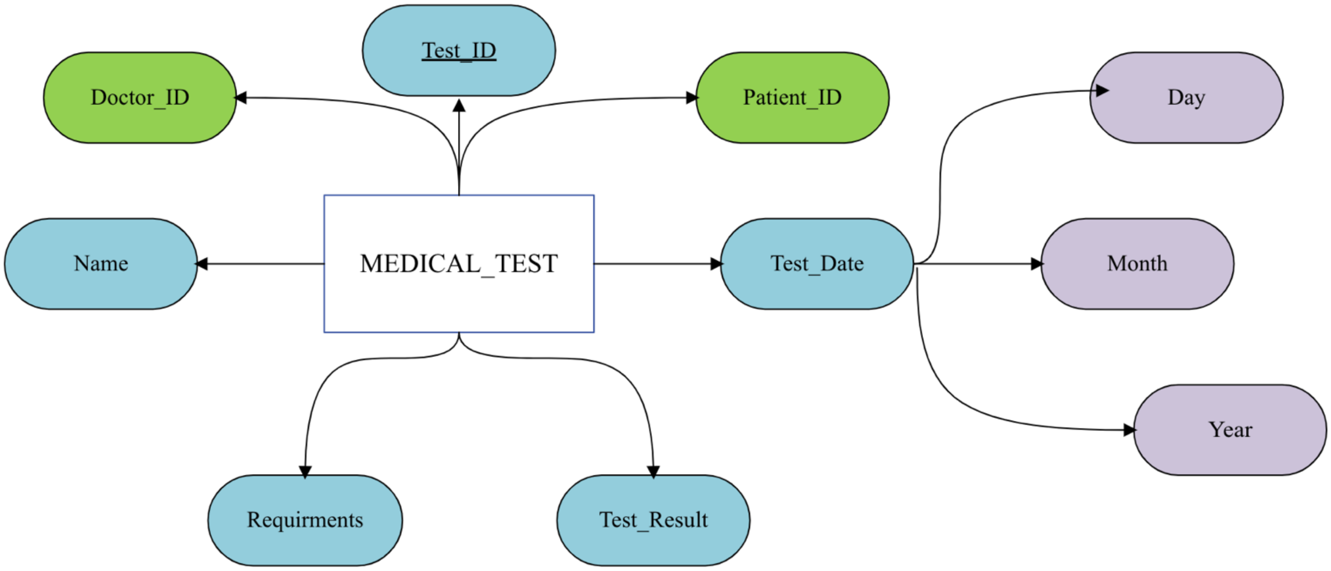
7. DIAGNOSIS



The **DIAGNOSIS** entity documents the diagnostic information, recording the condition identified and relevant details about the diagnosis. The primary key for **DIAGNOSIS** is **<u>Diagnosis_ID</u>**, a unique identifier for each diagnosis.

- Other Attributes:
  - **Condition_Name:** The name of the diagnosed condition.
  - **Description:** A detailed description of the condition and diagnosis.
  - **Diagnosis_Date:** The date the diagnosis was made, in the format MM/DD/YYYY.
- Foreign Keys:
  - **Patient_ID:** A reference to the patient receiving the diagnosis.
  - **Doctor_ID:** A reference to the doctor who made the diagnosis

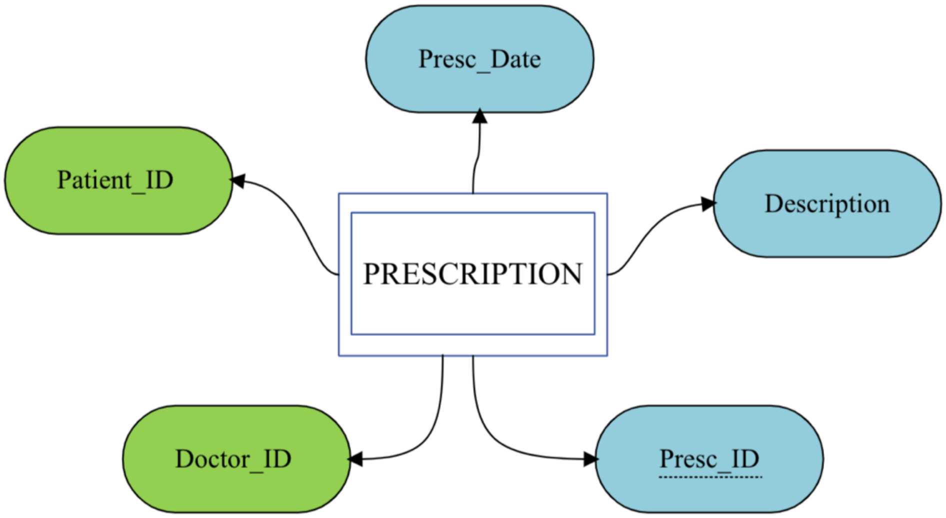
8. MEDICAL_TEST



This entity contains data on tests performed for diagnostic purposes, including results and any specific requirements for the test. The primary key for **MEDIACL_TEST** is **<u>Test_ID</u>**, a unique identifier for each medical test.

- Other Attributes:
  - **Name**: The name of the test (e.g., blood test, X-ray, MRI).
  - **Test_Result**: Stores the result of the test, which may vary based on the test type.
  - **Test_Date**:The date the test was conducted, formatted as MM/DD/YYYY.
  - **Requirements**: Lists any necessary preparations or conditions for conducting the test. (for example, for some medical tests, the patient should be fasting)
- Foreign Keys:
  - **Patient_ID**: A reference to the patient for whom the test is conducted.
  - **Doctor_ID**:A reference to the doctor who ordered or conducted the test.

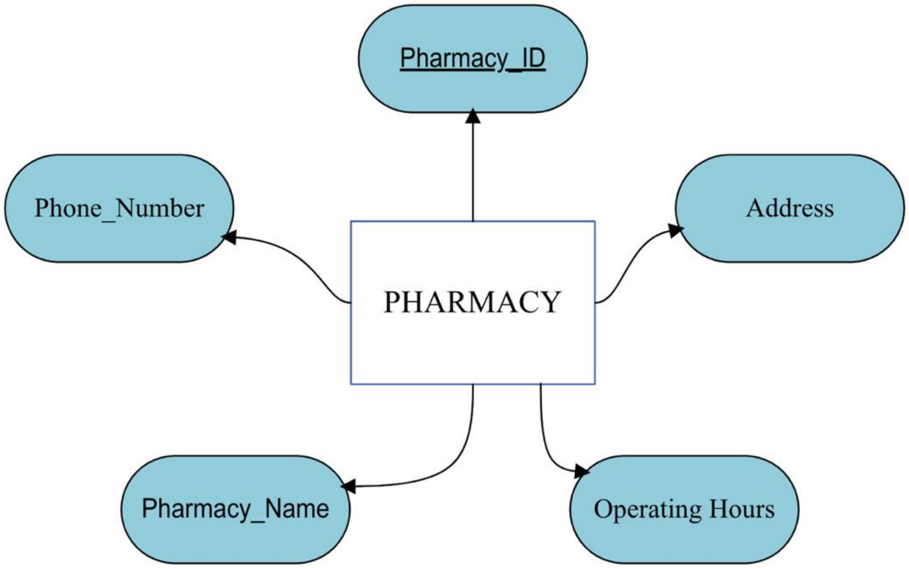
9. PRESCRIPTION



The **PRESCRIPTION** entity is a weak entity responsible for managing and tracking a patient's treatment plan. The **Pres_ID** is a partial key.

- Other Attributes:
  - **Presc_Date**: records when the prescription was issued, providing a clear timeline within the patient's medical history.
  - **Description**: provides detailed information about the treatment, including the medication name, treatment name, and any specific directions for administration.
- Foreign Keys:
  - **Patient_ID**: links the prescription to the specific patient receiving the treatment.
  - **Doctor_ID**: connects the prescription to the prescribing doctor, ensuring accountability and traceability.

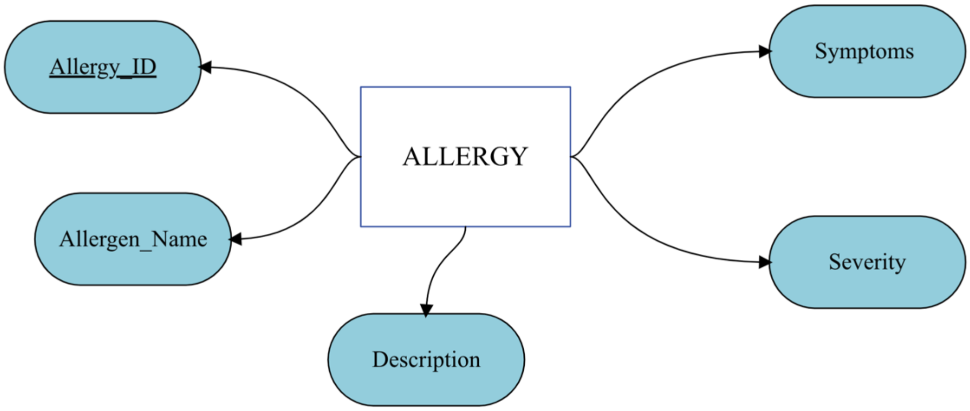
10. PHARMACY



The **PHARMACY** entity stores essential information about pharmacies that dispense medications to patients. This entity helps maintain an organized and accessible record of pharmacies within the healthcare system. Each pharmacy is uniquely identified by its **Pharmacy_ID**, which serves as the primary key.

- Other Attributes:
  - **Pharmacy_Name**: records the name of the pharmacy
  - **Phone_Number**: provides a contact number for quick communication.
  - **Address**: specifies the pharmacy's physical location, enabling patients and providers to locate it easily.
  - **Operating_Hours**: indicate the hours during which the pharmacy is open, ensuring that patients and healthcare providers are aware of when they can access pharmacy services.

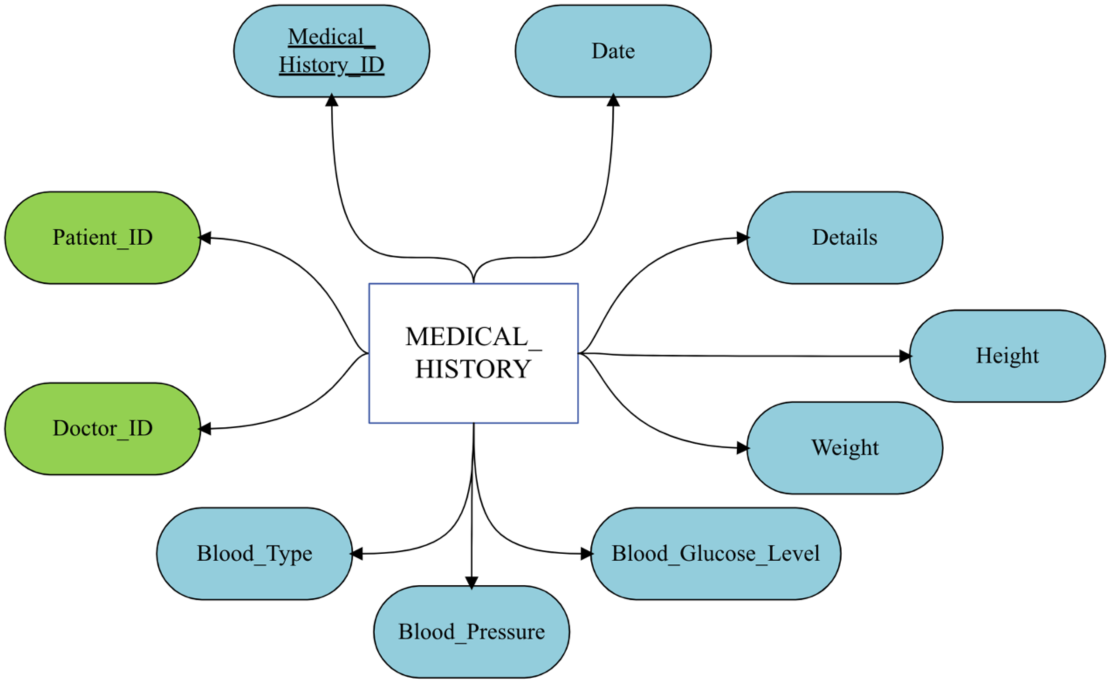
11. ALLERGY



Allergies can pose severe and life-threatening risks to individuals, making it essential to store detailed allergy information in our database. Knowing which allergies affect each patient is vital for effective treatment and ensures that healthcare providers can make informed, safe decisions for their care. The **ALLERGY** entity is identified by its unique **Allergy_ID**, which serves as a primary key.

- Other Attributes:
  - **Allergen_Name**: The name of the allergen (e.g., pollen, peanuts, latex), indicating what substance the patient is allergic to.
  - **Description**: A brief description of how the allergen impacts individuals, including the effects it may have on the human body.
  - **Severity**: The level of severity of the allergy (e.g., mild, moderate, severe), which helps healthcare providers prioritize and handle each case appropriately.
  - **Symptoms**: common symptoms experienced by the patient when exposed to the allergen, such as itching, swelling, or respiratory issues.

12. MEDICAL_HISTORY

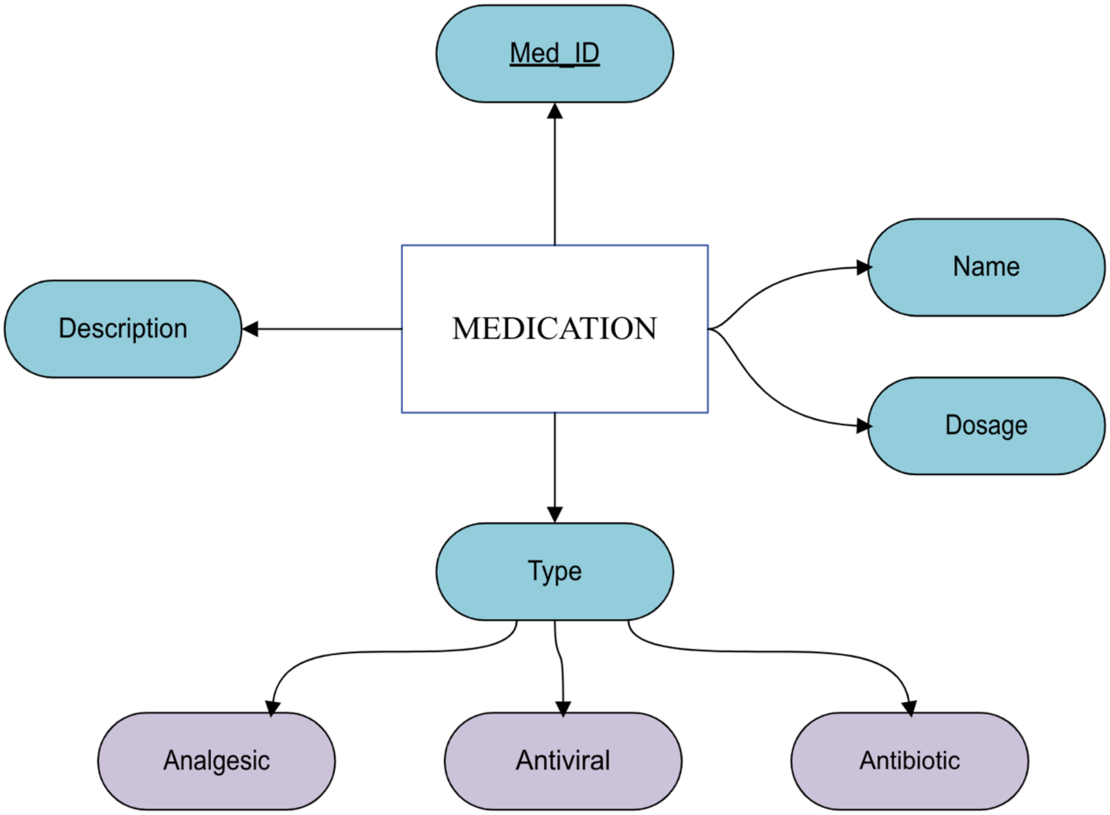


The **MEDICAL HISTORY** entity records a patient’s health history over time, including past medical conditions and treatments. This information supports patient care, allowing healthcare providers to reference previous health records (blood type, weight, height, details…) during treatment. The **MEDICAL_HISTORY**’s primary key is **Medical_History_ID**, a unique identifier for each medical history record.

- Other Attributes:
  - **Date**: The date the medical history entry was recorded in DD/MM/YYYY format.
  - **Details**: Notes on the patient’s past medical conditions, treatments, surgeries, or other relevant health information.
  - **Weight**: The patient’s weight, recorded at the time of the entry.
  - **Height**: The patient’s height, recorded at the time of the entry.
  - **Blood_Type**: The blood type of the patient (e.g., A+, O-).
  - **Blood_Pressure**: The recorded blood pressure at the time of the medical history entry.
  - **Blood_Glucose_Level**: The patient’s blood glucose level at the time of entry.

- Foreign Keys:
  - **Patient_ID**: A reference to the patient whose history is being recorded.
  - **Doctor_ID**: A reference to the doctor who created or updated the medical history record.

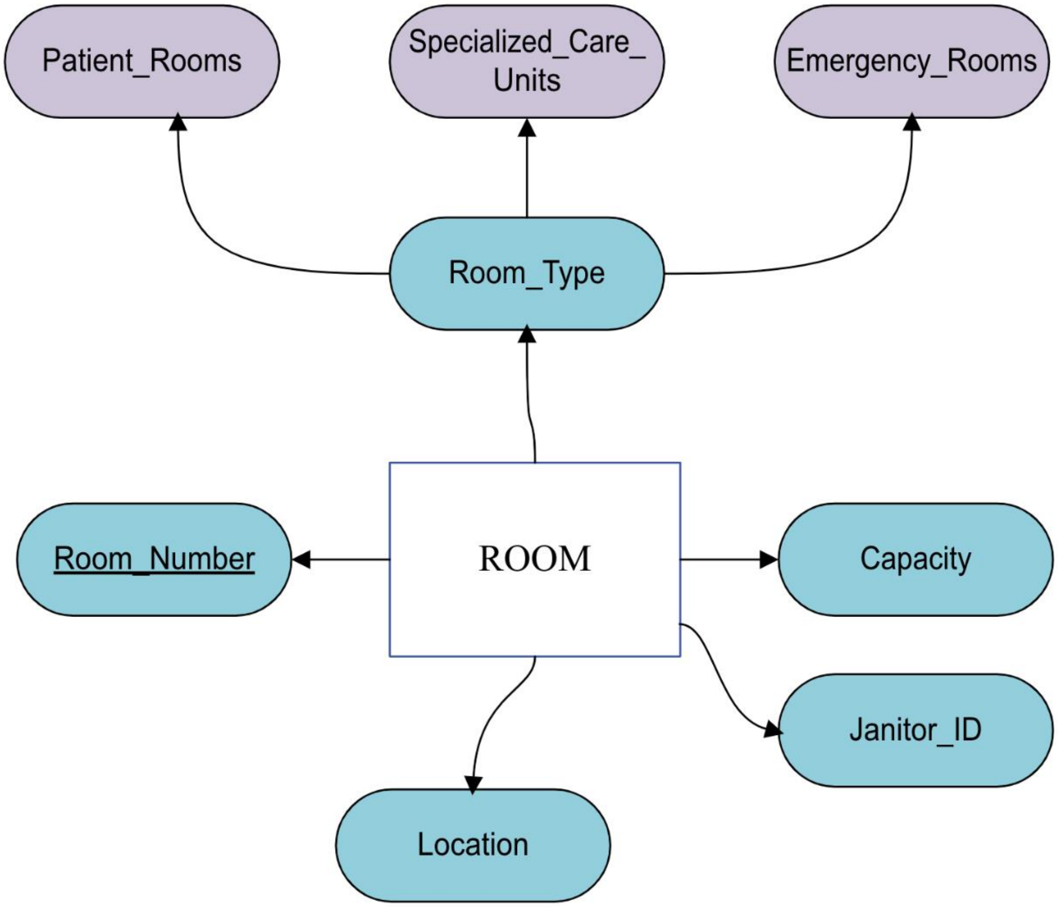
13. MEDICATION



The **MEDICATION** entity represents the medications that are available for prescription within our clinic. This entity includes details of each medication, including its name, type, usage, and prescribed dosage by the DOCTOR. The MEDICATION's primary key is **Med_ID**, a unique identifier for each medication.

- Other Attributes:
  - **Name**: The name of the medication.
  - **Description**: Details about the medication, such as its usage, instructions, or composition.
  - **Dosage**: The amount of the medication that should be taken according to the DOCTOR's prescription.
  - **Type**: The type or category of medication (e.g., antibiotic, analgesic, antiviral).

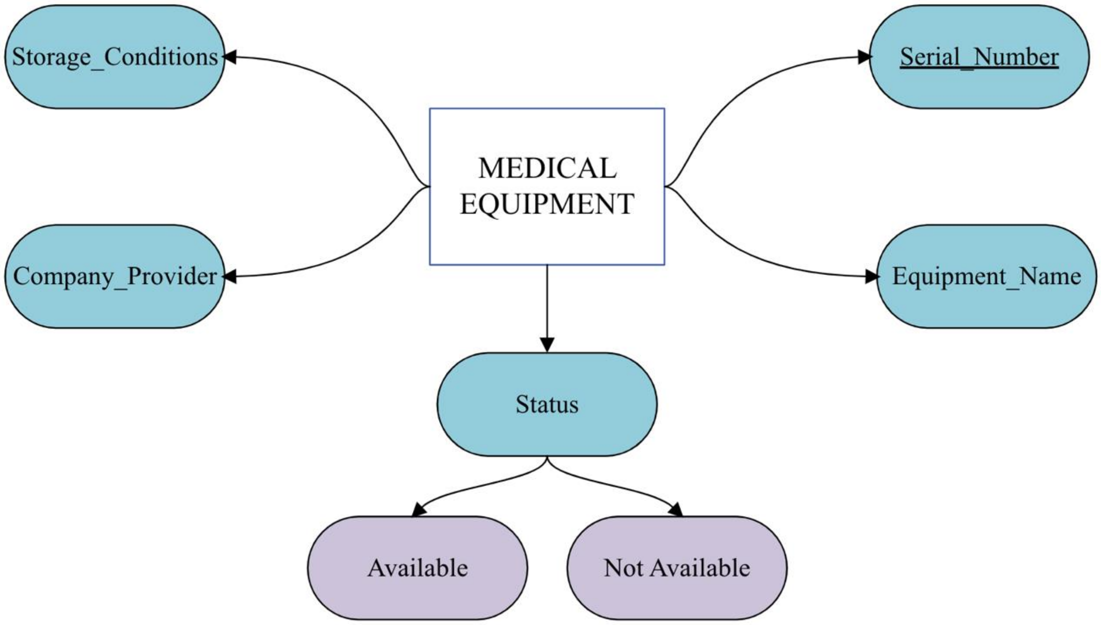
14. ROOM



The **ROOM** represents the place where our PATIENTs reside during their stay or their treatment. Each room has a unique identifier, a **<u>Room_Number</u>**, serving as the primary key.

- Other Attributes:
  - **Room_Type**: Specifies the type of room (e.g., private, semi-private, ICU).
  - **Capacity**: Indicates the maximum number of patients the room can support.
  - **Location**: The physical location of the room within the building or hospital.
  - **Janitor_ID**: Identifies the janitor responsible for maintaining this room.

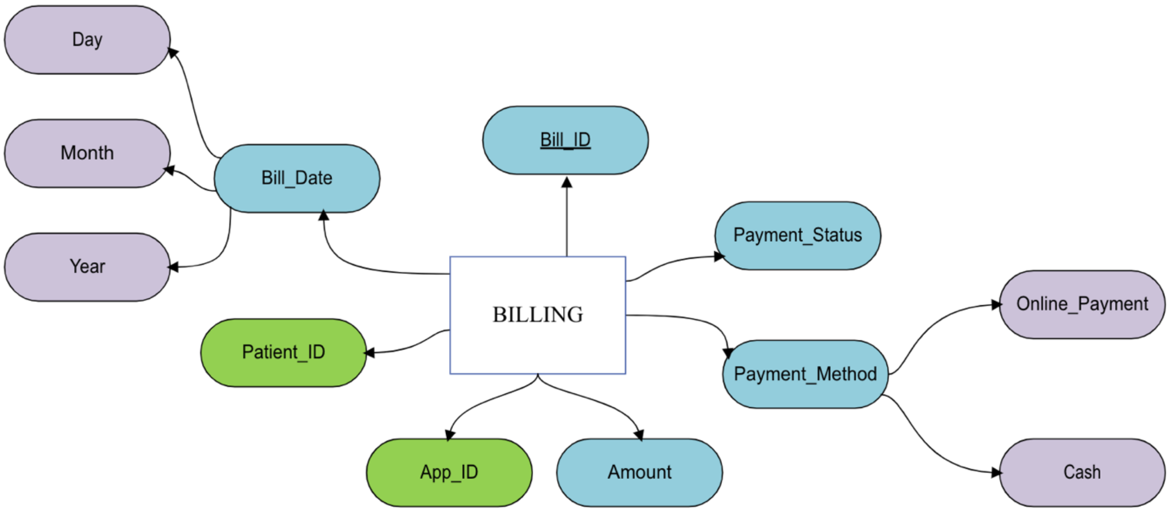
15. MEDICAL_EQUIPMENT



The **MEDICAL_EQUIPMENT** entity manages essential tools and devices used in medical procedures and patient care. It provides a catalog of available equipment with their status, as well as their storage conditions and their provider. The **MEDICAL_EQUIPMENT**'s primary key is the **Serial_Number**, a unique identifier for each piece of equipment.

- Other Attributes:
  - **Equipment_Name**: The name of the equipment, such as "Ultrasound Machine" or "Defibrillator."
  - **Status**: Indicates the availability of the equipment (e.g., available, in-use, under maintenance).
  - **Storage_Conditions**: Details specific requirements for storing the equipment safely, like temperature or humidity levels.
  - **Company_Provider**: The name of the company that supplied the equipment.

16. BILLING



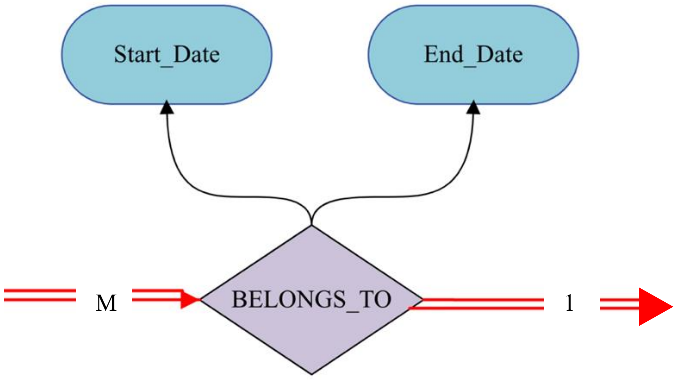
The **BILLING** entity is responsible for keeping track of our **PATIENT**s' bills. It is identified by a unique **Bill_ID,** serving as its primary key.

- Other Attributes:
  - **Bill_Date**: The date when the bill is generated.
  - **Amount**: Total amount due on the bill.
  - **Payment_Method**: Specifies the method of payment (e.g., cash, card, insurance).
  - **Payment_Status**: Indicates whether the bill is paid, partially paid, or unpaid.

- Foreign Keys:
  - **Patient_ID**: Links the bill to the associated patient in the PATIENT entity.
  - **App_ID**: Links the bill to the associated appointment in the APPOINTMENT entity.

VIII. Relationships:

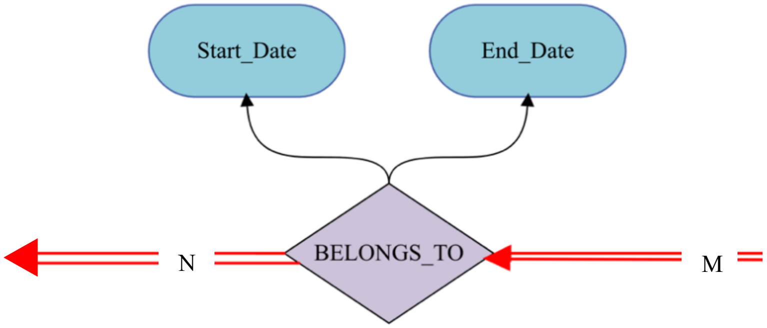
1. BELONGS_TO1



Each doctor should be associated with one department based on the doctor's expertise (e.g., Surgery, Ophthalmology, Pediatrics). Thus, a **"BELONGS TO1"** relationship must be established between the **DOCTOR** and the **DEPARTMENT**. The participation is total on both sides since each doctor should work in one department, and each department must have doctors assigned to it.

- A **DOCTOR "BELONGS TO1"** exactly one **DEPARTMENT**.
- A **DEPARTMENT** must contain at least one **DOCTOR**.
    - Attributes:
        - **Start_Date**: Stores the date when the doctor joined the department.
        - **End_Date**: Stores the date when the doctor left the department.

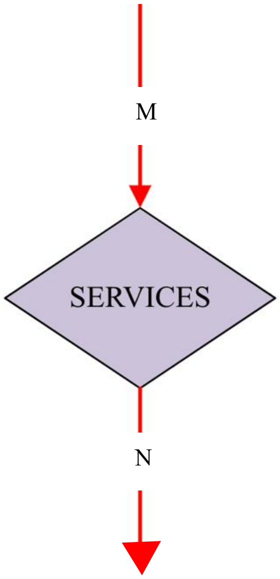
2. BELONGS_TO2



Each nurse should be associated with one department based on the nurse's expertise (e.g., Surgery, Ophthalmology, Pediatrics). Thus, a **"BELONGS TO2"** relationship must be created between the **NURSE** and the **DEPARTMENT**. The participation is total on both sides since each nurse should work in at least one department, and each department must have at least one nurse assigned to it.

- A **NURSE "BELONGS TO2"** at least one **DEPARTMENT**.
- A **DEPARTMENT** must contain at least one **NURSE**.
    - Attributes:
        - **Start_Date**: Stores the date when the nurse joined the department.
        - **End_Date**: Stores the date when the nurse left the department.

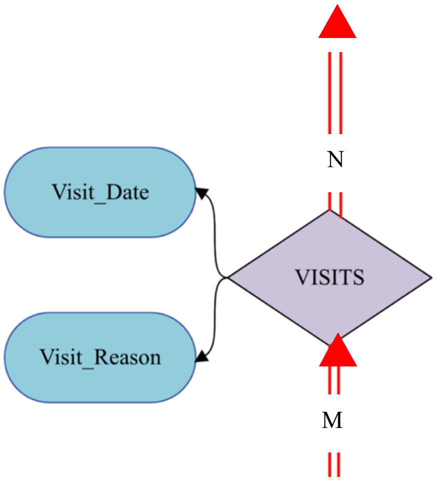
3. SERVICES



Many nurses take care of multiple patients. Thus, a "**SERVICES**" relationship must be created between the **NURSE** and the **PATIENT**. This participation is partial on both sides, since a nurse might not be assigned to any patient, and a patient might not need a nurse.

- Not every **NURSE "SERVICES"** a **PATIENT**.
- Not every **PATIENT** needs a **NURSE**.

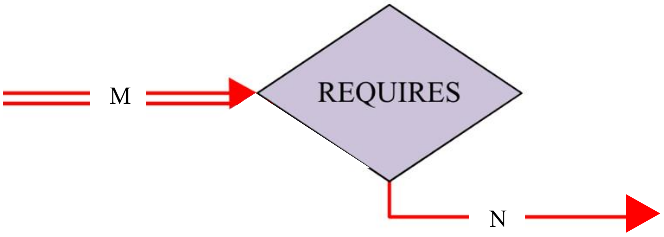
4. VISITS



The **VISITS** relationship documents instances of patients consulting with doctors. This relationship helps track patient-doctor interactions and supports scheduling, providing a record of each patient's healthcare visits. The participation is total on both sides, since each patient might visit one or more doctors, and a doctor might have one or more patients visiting them.

- A **PATIENT** "**VISITS**" one or more **DOCTOR** entries.
- A **DOCTOR** may have one or more **PATIENT** visits associated with them.
    - Attributes:
        - **Visit_Date**: Stores the date of the visit and time.
        - **Visit_Reason**: Stores the reason the patient is visiting the doctor.

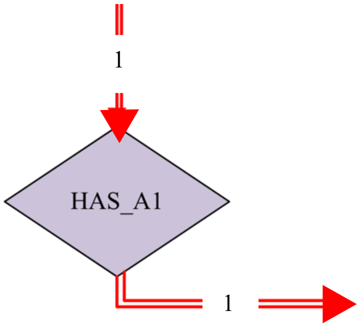
5. REQUIRES



A diagnosis may require one or more medical tests to confirm or further analyze the patient's condition. Therefore, a "**REQUIRES**" relationship is created between the **DIAGNOSIS** entity and the **MEDICAL_TEST** entity. Each diagnosis may require multiple medical test entries, while each medical test may be associated with one or more diagnoses as needed. The participation is total on the diagnosis side and partial on the medical test side.

- A **DIAGNOSIS "REQUIRES"** one or more **MEDICAL_TEST** records.
- A **MEDICAL_TEST** may be linked to one or more **DIAGNOSIS** records.

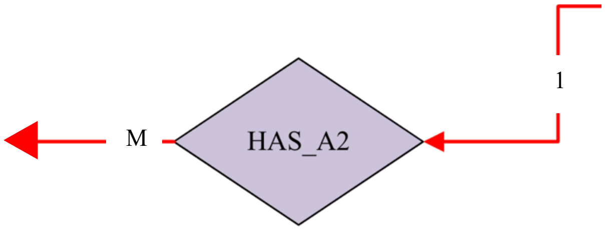
6. HAS A1



Each patient has a medical history that documents their health information over time. Therefore, a "**HAS A1**" relationship is established between the **Patient** entity and the **Medical History** entity. Each medical history entry must be associated with one patient. The participation on both sides is total.

- A **PATIENT "HAS A1"** one **MEDICAL_HISTORY** record.
- A **MEDICAL_HISTORY** record belongs to exactly one **PATIENT**.

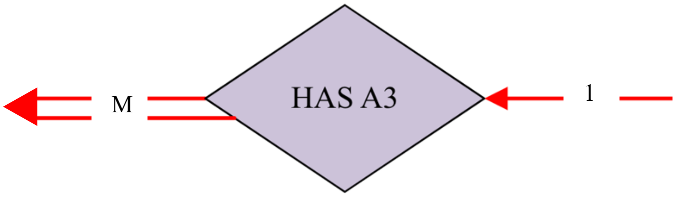
7. HAS A2



Patients may have documented allergies to ensure safe medical treatments. A "**HAS A2**" relationship exists between the **PATIENT** entity and the **ALLERGY** entity. This participation is partial on both sides, as not every patient has an allergy, and allergies must be documented individually for each patient.

- A **PATIENT "HAS A2"** one or more **ALLERGY** records.
- An **ALLERGY** record belongs to exactly one **PATIENT**.

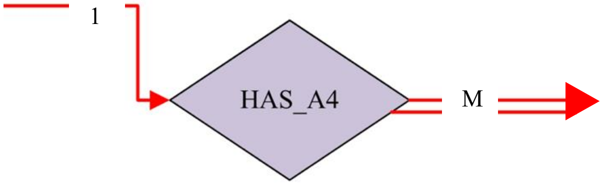
8. HAS A3



Patients may have booked an appointment with a doctor for consultation, regular checkup, or other health issues. Thus, a "**HAS A3**" relationship must exist between the **PATIENT** and the **APPOINTMENT**. This participation is partial on the patient's side and total on the appointment's side.

- A **PATIENT "HAS A3" APPOINTMENT**.
- An **APPOINTMENT** must be linked to a **PATIENT**.

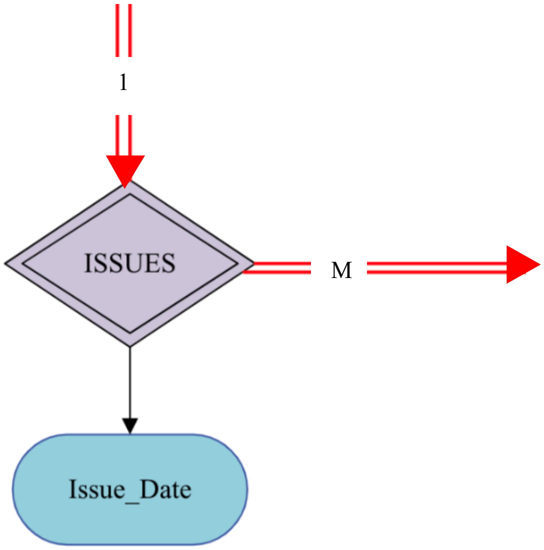
9. HAS A4



Each patient can have one or more diagnoses documented for medical purposes. Thus, a "**HAS A4**" relationship exists between the **PATIENT** entity and the **DIAGNOSIS** entity. The participation is partial on the patient side, as not every patient has a diagnosis, and total on the diagnosis side, since each diagnosis entry must be linked to a patient.

- A **PATIENT "HAS A4"** one or more **DIAGNOSIS** records.
- A **DIAGNOSIS** record belongs to exactly one **PATIENT**.

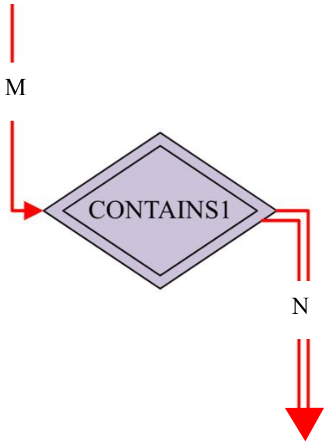
10. ISSUES



The "**ISSUES"** relationship is a weak relationship linking doctors to prescriptions they create for patients. This relationship captures the doctor's authority in prescribing medications, providing a connection between the healthcare provider and the prescribed treatment. The participation is total on both sides, since each doctor may issue a prescription, and each prescription should be issued by a doctor.

- A **DOCTOR** "**ISSUES**" one or more **PRESCRIPTION** records.
- A **PRESCRIPTION** is associated with one **DOCTOR** who issues it.
  - Attributes:
    - **Issue_Date**: Stores the date on which the doctor issued the prescription.

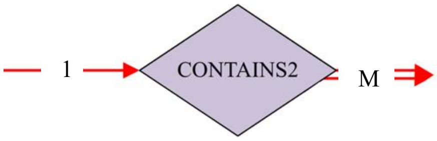
11. CONTAINS1



Each prescription may contain one or more medications necessary for a patient's treatment. Thus, a "**CONTAINS1**" relationship must be established between the **PRESCRIPTION** entity and the **MEDICATION** entity. The participation is partial on the prescription side, as not every prescription will contain medications (e.g., some may involve non-medication treatments), while it is total on the medication side, since each medication must be associated with at least one prescription to ensure proper usage.

- Not every **PRESCRIPTION "CONTAINS1" MEDICATION**.
- A **MEDICATION** requires the presence of a **PRESCRIPTION**.

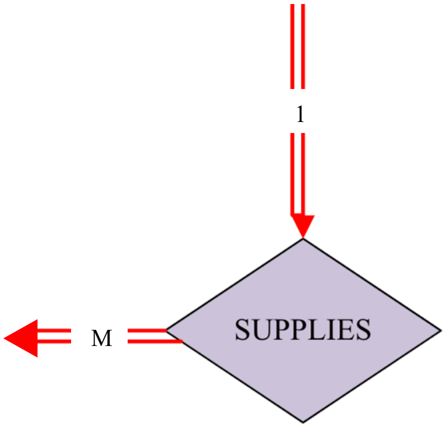
12. CONTAINS2



A room might contain medical equipment needed for a patient's examination or treatment. Thus, a "**CONTAINS2**" relationship must be established between the **ROOM** entity and the **MEDICAL_EQUIPMENT** entity. The participation is partial on the room's side, as not all rooms should contain medical equipment, while it is total on the medical equipment's side, since each medical equipment should be assigned or stored in a room.

- Not every **ROOM "CONTAINS2" MEDICAL_EQUIPMENT** (e.g. waiting room).
- A **MEDICAL_EQUIPMENT** must be assigned to a **ROOM** (e.g. storage room, surgery room).

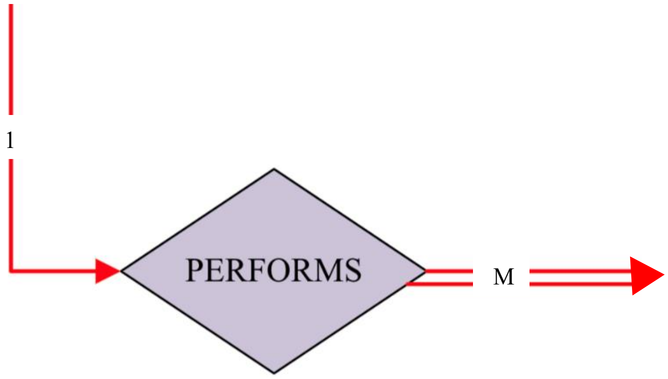
13. SUPPLIES



A pharmacy will supply our clinic with the medication needed to treat patients. Thus, a "**SUPPLIES**" relationship must be established between the **PHARMACY** entity and the **MEDICATION** entity. The participation is total on both sides, as the pharmacy must supply at least one medication, and the medication used in the clinic must be supplied by the pharmacy.

- A **PHARMACY "SUPPLIES" MEDICATIONS**.
- A **MEDICATION** must be supplied by the **PHARMACY**.

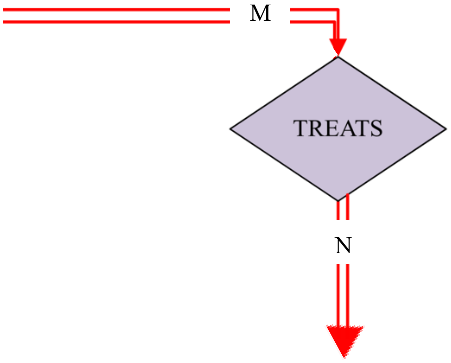
14. PERFORMS



The **PERFORMS** relationship links doctors to the treatments they administer to patients. It documents each treatment procedure and the healthcare provider responsible, ensuring a complete record of medical care provided. The participation on the doctor's side is partial since not all doctors might perform a treatment (e.g., regular health checkups), and total on the treatment side, as every treatment must have a doctor responsible for performing it.

- Not every **DOCTOR** "**PERFORMS**" **TREATMENT** procedures.
- Each **TREATMENT** entry must be associated with one **DOCTOR**.

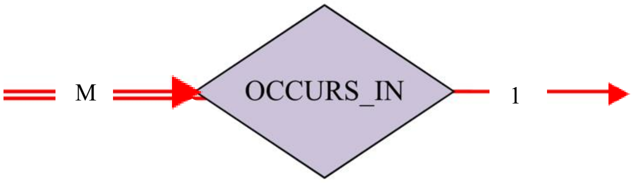
15. TREATS



Every patient requires treatment from a doctor; thus, a "**TREATS**" relationship is established between the Doctor and Patient entities. The participation is total on both sides, as each doctor must treat one or more patients, and each patient must be treated by at least one doctor.

- A **DOCTOR "TREATS"** one or more **PATIENTS**.
- A **PATIENT** may be treated by one or more **DOCTORS**.

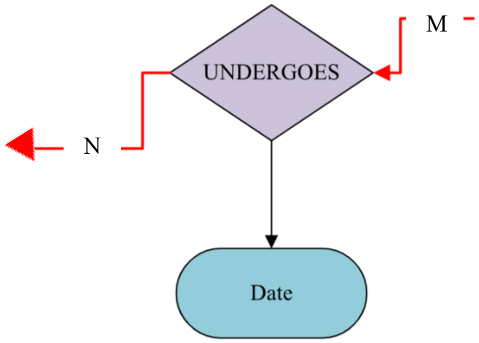
16. OCCURS_IN



Each treatment is conducted within a specific room, necessitating the establishment of an "**OCCURS_IN**" relationship between the **TREATMENT** entity and the **ROOM** entity. The participation is total on the treatment side because every treatment must take place in a designated room, ensuring that proper facilities and equipment are available. On the room side, participation is partial, as not every room may be utilized for treatment purposes—some rooms may be reserved for other functions, such as storage or administrative tasks.

- A **TREATMENT "OCCURS_IN"** only one **ROOM** at a time.
- A **ROOM** may accommodate multiple **TREATMENTs** over time.

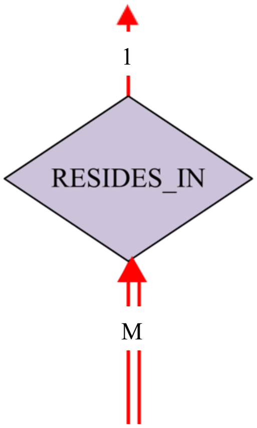
17. UNDERGOES



The **UNDERGOES** relationship records the treatments patients receive. It helps maintain a comprehensive medical history for each patient, connecting them to treatments undertaken for their healthcare. The participation for both sides is partial since not all patients might undergo treatments and not all treatments must be associated with a patient.

- A **PATIENT** can "**UNDERGOES**" one or more **TREATMENT** entries.
- A **TREATMENT** must be associated with one or more **PATIENT** entries.
    - Attributes:
        - **Date**: Stores the date when the patient began the treatment.

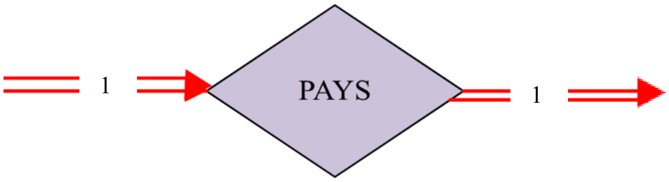
18. RESIDES_IN



Each patient requires a designated room during their stay, forming a "**RESIDES_IN**" relationship between the Patient and Room entities. Participation is total on the patient's side, as each patient must reside in exactly one room, and partial on the room's side, since some rooms might not be occupied by patients.

- A **PATIENT "RESIDES_IN"** only one **ROOM**.
- A **ROOM** may accommodate one or more **PATIENTs**.

19. PAYS



Patients are responsible for settling their bills, so a “**PAYS**” relationship is created between the Patient and Bill entities. Participation is total on both sides, as each patient must pay a bill, and each bill must be paid by a patient.

- A **PATIENT “PAYS**” one **BILL**.
- A **BILL** must be paid by one **PATIENT**.

IX. ER to Relational Mapping Algorithm:

After designing the ER schema and displaying the database for The Clinic of Lost Souls as a system of entities, attributes, and relationships, this high-level design must be translated into a relational database design. To map the ER design to a relational database design, a seven-step algorithm needs to be followed. Relational database design (Elmasri, 2010) ultimately produces a set of relations. The implicit goals of the design activity are information preservation and minimum redundancy. The following is a detailed description of applying the different steps to our database design. The steps, in brief, are as follows:

Step 1: All regular entity types are mapped into relations schemas. By regular, we mean that only non-weak entities will be mapped in this step. For every regular entity, only the simple attributes are encoded into the relation schemas. Composite attributes are broken down into their simple attribute components. Multivalued and derived attributes are not encoded in this step. Multivalued attributes will be added in the Step 6.

Step 2: All weak entity types are mapped into relation schemas. As in Step 1, only the simple attributes are encoded into the relation schemas. Composite attributes are broken down into their simple attribute components. Multivalued and derived attributes are not encoded in this step. Multivalued attributes will be added in the Step 6.

Step 3: All binary 1:1 relationship types are mapped into relation schemas. Specifically, in this step, we apply the foreign key approach where we choose the entity on the total participation side of the relation, then we add as a foreign key the primary key of the other entity participating in this relation.

Step 4: All binary 1:N relationship types are mapped into relation schemas. As in Step 3, we apply the foreign key approach. We add a foreign key in the entity type at the many sides of the relationship. This foreign key will be the primary key of the other entity type participating in this relationship.

Step 5: All binary M: N relationship types are mapped into relation schemas. Unlike Steps 3 and 4, we encode the relationships by creating a new relation that includes, as foreign keys, the primary keys of all participating relations. Their combination will form the primary key to this newly created relation.

Step 6: All multivalued attributes that were left over from the previous steps are mapped. Specifically, a relation is created for every multivalued attribute. It will contain the primary key of each entity that has a multivalued attribute.

Step 7: All N-ary relationship types are mapped in this step. A new relation containing the primary keys of all participating entities and any simple attributes of the relationship type is created. In our design, we have no N-ary relationship types, so this step is not applicable here.

STEP 1: Mapping of Regular Entity Types

In the first step, the regular entity types must be mapped into relations. Each regular entity is going to have its own relation that includes all of its simple attributes and a single primary key, which is underlined. The regular (strong) entities in this database design are: **DOCTOR**, **NURSE**, **PATIENT**, **DEPARTEMENT**, **APPOINTMENT**, **TREATMENT**, **DIAGNOSIS**, **MEDICAL_TEST**, **PHARMACY**, **ALLERGY**, **MEDICAL_HISTORY**, **MEDICATION**, **ROOM**, **MEDICAL_EQUIPMENT**, **BILLING**.

1. DOCTOR

ID	First_Name	Middle_Name	Last_Name	Email	Address	DOB	Gender	Specialization	Salary

The **DOCTOR** entity contains the simple attributes only. This relation only includes all simple attributes and the primary key **ID**. This entity has a composite attribute **Name** broken down into simple attributes: **First_Name**, **Middle_Name**, and **Last_Name**. The **DOCTOR** entity is also composed of other attributes such as:

- **First_Name**
- **Middle_Name**
- **Last_Name**
- **Email**
- **Address**
- **DOB**
- **Gender**
- **Specialization**
- **Salary**

2. NURSE

ID	First_Name	Middle_Name	Last_Name	Email	Address	Specialization	DOB	Gender	Shift	Salary

The **NURSE** entity contains the simple attributes only. This relation only includes all simple attributes and the primary key **ID**. This entity has a composite attribute **Name** broken down into simple attributes: **First_Name**, **Middle_Name**, and **Last_Name**. The **NURSE** entity is also composed of other attributes such as:

- **First_Name**
- **Middle_Name**
- **Last_Name**
- **Email**
- **Address**
- **Specialization**

- **DOB**
- **Gender**
- **Shift**
- **Salary**

3. PATIENT

ID	First_Name	Middle_Name	Last_Name	Email	Address	DOB	Emergency_Contact	Gender

The **PATIENT** entity contains the simple attributes only. This relation only includes all simple attributes and the primary key **ID**. This entity has a composite attribute **Name** broken down into simple attributes: **First_Name**, **Middle_Name**, and **Last_Name**. The **PATIENT** entity is also composed of other attributes such as:

- **First_Name**
- **Middle_Name**
- **Last_Name**
- **Email**
- **Address**
- **DOB**
- **Emergency_Contact**
- **Gender**

4. DEPARTMENT

Department_ID	Department_Name

The **DEPARTMENT** entity contains only the simple attributes. This relation only includes all simple attributes and the primary key **Department_ID**. The **DEPARTMENT** entity also has one other attribute, which is **Department_Name**.

5. APPOINTMENT

Appointment_Nb	App_Date	Time

The **APPOINTMENT** entity contains the simple attributes only. This relation only includes all simple attributes and the primary key **Appointment_Nb**. The **APPOINTMENT** entity is also composed of other attributes such as:

- **App_Date**
- **Time**

6. TREATMENT

Treatment_ID	Type	Procedure	Requirements

The **TREATMENT** entity contains the simple attributes only. This relation only includes all simple attributes and the primary key **Treatment_ID**. The **TREATMENT** entity is also composed of other attributes such as:

- **Type**
- **Procedure**
- **Requirements**

7. DIAGNOSIS

Diagnosis_ID	Condition_Name	Description	Diagnosis_Date

The **DIAGNOSIS** entity contains the simple attributes only. This relation only includes all simple attributes and the primary key **Diagnosis_ID.** The **DIAGNOSIS** entity is also composed of other attributes such as:

- **Condition_Name**
- **Description**
- **Diagnosis_Date**

8. MEDICAL_TEST

Test_ID	Name	Test_Result	Test_Date	Requirements

The **MEDICAL_TEST** entity contains the simple attributes only. This relation only includes all simple attributes and the primary key **Test_ID** The **MEDICAL_TEST** entity is also composed of other attributes such as:

- **Name**
- **Test_Result**
- **Test_Date**
- **Requirements**

9. PHARMACY

Pharmacy_ID	Phone_Number	Pharmacy_Name	Address	Operating_Hours

The **PHARMACY** entity contains the simple attributes only. This relation only includes all simple attributes and the primary key **Pharmacy_ID**. The **PHARMACY** entity is also composed of other attributes such as:

- **Phone_Number**
- **Pharmacy_Name**
- **Address**
- **Operating Hours**

10. ALLERGY

Allergy_ID	Allergen_Name	Description	Symptoms	Severity

The **ALLERGY** entity contains the simple attributes only. This relation only includes all simple attributes and the primary key **Allergy_ID**. The **ALLERGY** entity is also composed of other attributes such as:

- **Allergen_Name**
- **Description**
- **Symptoms**
- **Severity**

11. MEDICAL_HISTORY

Medical_History_ID	Date	Details	Weight	Height	Blood_Type	Blood_Pressure	Blood_Glucose_Level

The **MEDICAL_HISTORY** entity includes only simple attributes and the primary key **Medical_History_ID**. The **MEDICAL_HISTORY** entity is also composed of other attributes such as:

- **Date**
- **Details**
- **Weight**
- **Height**
- **Blood_Type**
- **Blood_Pressure**

- **Blood_Glucose_Level**

12. MEDICATION

Med_ID	Name	Description	Type	Dosage

The **MEDICATION** entity contains the simple attributes only. This relation only includes all simple attributes and the primary key **Med_ID**. The **MEDICATION** entity is also composed of other attributes such as:

- **Name**
- **Description**
- **Type**
- **Dosage**

13. ROOM

Room_Number	Room type	Capacity	Janitor_ID	Location

The **ROOM** entity contains the simple attributes only. This relation only includes all simple attributes and the primary key **Room_Number**. The **ROOM** entity is also composed of other attributes such as:

- **Room_type**
- **Capacity**
- **Janitor_ID**
- **Location**

14. MEDICAL_EQUIPMENT

Serial_Number	Equipment_Name	Status	Storage_Conditions	Company_Provider

The **MEDICAL_EQUIPMENT** entity contains the simple attributes only. This relation only includes all simple attributes and the primary key **Serial_Number**. The **MEDICAL_EQUIPMENT** entity is also composed of other attributes such as:

- **Equipment_Name**
- **Status**
- **Storage_Conditions**
- **Company_Provider**

15. BILLING

Bill_ID	Bill_Date	Amount	Payment_Method	Payment_Status

The **BILLING** entity contains the simple attributes only. This relation only includes all simple attributes and the primary key **Bill_ID**. The **BILLING** entity is also composed of other attributes such as:

- **Bill_Date**
- **Amount**
- **Payment_Method**
- **Payment_Status**

STEP 2: Mapping of Weak Entity Types

In this step, the weak entity types are mapped into relations. As in Step 1, only the simple attributes are included in the relations and not multivalued or derived attributes. Furthermore, weak entity relation has a foreign key attribute, the primary key of the owner entity type. The combination of the foreign key added, and the weak entity type's partial key represents the relation's primary key.
The weak entity in our database design is **PRESCRIPTION**.

1. PRESCRIPTION

Presc_ID	Doctor_ID	Patient_ID	Description	Presc_Date

The weak entity in our database design is **PRESCRIPTION**. The simple attributes included are Presc_Date and Description. The primary key of the PATIENT entity, **ID**, is renamed as **Patient_ID**, whereas the primary key of the **DOCTOR** entity, **ID**, is renamed as **Doctor_ID.** Additionally, the **Patient_ID** and **Doctor_ID** are included in this relation. The combination of **Patient_ID, Doctor_ID,** and the partial key **Pres_ID** represents the primary key of this relation.

STEP 3: Mapping of Binary 1:1 Relationship Types

In this step, we are going to map the binary one-to-one relationship. To accomplish our goal, we can follow one of three approaches. The first approach called the foreign key approach, is where we choose the entity on the total participation side of the relation, and then we add, as a foreign key, the primary key of the other entity participating in this relation. The second approach called the merged relation approach, is where we merge the two entities participating in the relationship into a single relation. This is only used when both participations are total. The third approach, called the cross-reference or relationship relation approach, is where we create a third relation, which will include the primary keys of both entities participating in the relationship. We are going to follow the foreign key approach. The binary one-to-one relationships that need to be mapped are: **HAS_A1** and **PAYS**.

1. MEDICAL_HISTORY (HAS A1)

Medical_History_ID	Patient_ID	Date	Details	Height	Weight	Blood_Glucose_Level	Blood_Pressure
Blood_Type	Doctor_ID						

The “**HAS_A1**” relationship links the **PATIENT** and **MEDICAL_HISTORY** entities. The participation is total on both sides. Thus, it does not matter where we add the foreign key that relates to both entities. This is a one-to-one relationship, where we chose the **MEDICAL_HOSTORY** relation in which we added, as a foreign key, the primary key **ID** of the patient and renamed it as **Patient_ID**.

2. BILLING (PAYS)

Bill_ID	Patient_ID	Bill_Date	Amount	Payment_Status	Payment_Method	App_ID

The “**PAYS**” relationship links the **PATIENT** and **BILLING** entities. The participation is total on both sides. Thus, it does not matter where we add the foreign key that relates to both entities. This is a one-to-one relationship, where we chose the **BILLING** relation in which we added, as a foreign key, the primary key **ID** of the patient and renamed it as **Patient_ID**.

STEP 4: Mapping of Binary 1:N Relationship Types

In this step, we are going to map the binary one to many relationships. We add a foreign key in the entity type at the many sides of the relationship. This foreign key is the primary key of the other entity type participating in this relationship. We must also include any other simple attribute of the one-to-many relationship. The one-to-many relationships that need to be mapped are: **BELONGS_TO1**, **HAS_A2**, **HAS_A3**, **HAS_A4**, **ISSUES**, **CONTAINS2**, **SUPPLIES**, **PERFORMS**, **OCCURS_IN**, and **RESIDES_IN**.

1. DOCTOR (BELONGS_TO1)

ID	Department_ID	First_Name	Middle_Name	Last_Name	Email	Address	DOB	Gender
Specialization	Salary							

Many doctors belong to one department. Thus, the "**BELONGS_TO1**" relationship links the **DOCTOR** entity and the **DEPARTMENT** entity. In this relationship, the **DOCTOR** entity is on the "many" side. As a result, a foreign key, **Department_ID**, which is the primary key of the **DEPARTMENT** entity, is added to this relation.

2. ALLERGY (HAS_A2)

Allergy_ID	Patient_ID	Allergen_Name	Description	Symptoms	Severity

Many allergies might belong to one patient. Thus, the "**HAS_A2**" relationship links the **PATIENT** entity and the **ALLERGY** entity. In this relationship, the **ALLERGY** entity is on the "many" side. As a result, a foreign key, **ID**, which is the primary key of the **PATIENT** entity, is renamed as **Patient_ID** and added to this relation.

3. APPOINTMENT (HAS_A3)

Appointment_Nb	Patient_ID	Day	Month	Year	Hour	Minute	Doctor_ID

A patient can have multiple appointments. Thus, the "**HAS_A3**" relationship links the **PATIENT** entity and the **APPOINTMENT** entity. In this relationship, the **APPOINTMENT** entity is on the "many" side. As a result, a foreign key, **ID** renamed as **Patient_ID,** which is the primary key of the **PATIENT** entity, is added to this relation.

4. DIAGNOSIS (HAS_A4)

Diagnosis_ID	Patient_ID	Condition_Name	Description	Doctor_ID	Diagnosis_Date

Many diagnoses might belong to one patient. Thus, the "**HAS_A4**" relationship links the **PATIENT** entity and the **DIAGNOSIS** entity. In this relationship, the **DIAGNOSIS** entity is on the "many" side. As a result, a foreign key, **ID** renamed as **Patient_ID**, which is the primary key of the **PATIENT** entity, is added to this relation.

5. PRESCRIPTION (ISSUES)

Presc_ID	Doctor_ID	Presc_Date	Patient_ID	Description	Issue_Date

A doctor can issue multiple prescriptions. Thus, the "**ISSUES**" relationship links the **DOCTOR** entity and the **PRESCRIPTION** entity. In this relationship, the **PRESCRIPTION** entity is on the "many" side. As a result, a foreign key, **ID**, which is the primary key of the **DOCTOR** entity, is renamed as **Doctor_ID** and added to this relation. The combination of **Presc_ID** and **Doctor_ID** represents the primary key of this relation.

6. MEDICAL_EQUIPMENT (CONTAINS2)

Serial_Number	Room_Number	Equipment_Name	Status	Company_Provider	Storage_Conditions

A room might contain several medical equipment. Thus, the "**CONTAINS2**" relationship links the **ROOM** entity and the **MEDICAL_EQUIPMENT** entity. In this relationship, the **MEDICAL_EQUIPMENT** entity is on the "many" side. As a result, a foreign key, **Room_Number**, which is the primary key of the **ROOM** entity, is added to this relation.

7. MEDICATION (SUPPLIES)

Med_ID	Pharmacy_ID	Name	Dosage	Type	Description

A pharmacy supplies medication. Thus, the "**SUPPLIES**" relationship links the **PHARMACY** entity and the **MEDICATION** entity. In this relationship, the **MEDICATION** entity is on the "many" side. As a result, we create a new relation to handle this one-to-many relationship. This new relation, "**SUPPLIES"**, will include foreign key **Pharmacy_ID**, the primary key of the **PHARMACY** entity.

8. TREATMENT (PERFORMS)

Treatment_ID	Doctor_ID	Type	Procedure	Patient_ID	Requirements

The “**PERFORMS**” relationship links the **DOCTOR** and **TREATMENT** entities. A doctor can perform multiple treatments, but each treatment is performed by one doctor. This is a one-to-many relationship, where the **TREATMENT** entity is on the many side. As a result, a foreign key, **ID**, which is the primary key of the **DOCTOR** entity, is renamed as **Doctor_ID** and added to the relation.

9. TREATMENT (OCCURS_IN)

Treatment_ID	Room_Number	Type	Procedure	Patient_ID	Doctor_ID

Many treatments occur in a room. Thus, the **OCCURS_IN** relationship links the **ROOM** entity and the **TREATMENT** entity. In this relationship, the **TREATMENT** entity is on the “many” side. As a result, a foreign key, **Room_Number,** which is the primary key of the **ROOM** entity, is added to this relation.

10. PATIENT (RESIDES_IN)

ID	Room_Number	First_Name	Middle_Name	Last_Name	Email
Address	DOB	Gender	Emergency_Contact		

The **RESIDES_IN** relationship links the **PATIENT** and **ROOM** entities. A patient resides in a single room, but a room can house multiple patients. This is a one-to-many relationship with the **PATIENT** entity on the “many” side. As a result, a foreign key, **Room_Number,** which is the primary key of the **ROOM** entity, is added to this relation.

STEP 5: Mapping of Binary M:N Relationship Types

In this step, we will map the binary many to many relationships. For each many-to-many relationship, we are going to create a new relation that includes, as foreign keys, the primary keys of all participating relations. Their combination will form the primary key of this newly created relation. We must also include any other simple attributes of the many-to-many relationship. The many-to-many relationships that need to be mapped are **BELONGS_TO2**, **SERVICES**, **VISITS**, **REQUIRES**, **CONTAINS1**, **TREATS**, and **UNDERGOES**.

1. BELONGS_TO2

Nurse_ID	Department_ID	Start_Date	End_Date

Many nurses belong to many departments. Thus, the "**BELONGS_TO2**" relationship is needed to link the **NURSE** entity and the **DEPARTMENT** entity. In this relationship, the **NURSE** entity and the **DEPARTMENT** are both on the "many" side. As a result, foreign keys, **ID** and **Department_ID**, which are the primary keys of the **NURSE** and **DEPARTMENT** entities respectively, are added to it. The "**BELONGS_TO2**" relationship has two attributes, **Start_Date** and **End_Date**, which are included in this relation. The combination of **ID** and **Department_ID** represents the primary key of the "**BELONGS_TO2**" relation.

2. SERVICES

Nurse_ID	Patient_ID

Many nurses service many patients. Thus, the "**SERVICES**" relationship is needed to link the **NURSE** entity and the **PATIENT** entity. In this relationship, the primary keys of the **NURSE** and **PATIENT** entities are included. The primary key of the **NURSE** entity, **ID**, is added to the "**SERVICES**" relation and renamed **Nurse_ID**. Also, the primary key of the **PATIENT** entity, **ID**, is added as **Patient_ID**. The combination of both keys represents the primary key of the "**SERVICES**" relation.

3. VISITS

Patient_ID	Doctor_ID	Visit_Date	Visit_Reason

Many patients visit many doctors. Thus, the "**VISITS**" relationship is needed to link the **PATIENT** entity and the **DOCTOR** entity. In this relationship, the primary keys of the **PATIENT** and **DOCTOR** entities are included. The primary key of the **PATIENT** entity, **Patient_ID**, is added to the "**VISITS**" relation. Also, the primary key of the

DOCTOR entity, **Doctor_ID**, is added. The "**VISITS**" relationship has two attributes, **Visit_Date** and **Visit_Reason**, which are included in this relation. The combination of **Patient_ID** and **Doctor_ID** represents the primary key of the "**VISITS**" relation.

4. REQUIRES

Test_ID	Diagnosis_ID

Many diagnoses require many tests. Thus, the "**REQUIRES**" relationship is needed to link the **DIAGNOSIS** entity and the **MEDICAL_TEST** entity. In this relationship, the primary keys of the **DIAGNOSIS** and **MEDICAL_TEST** entities are included. The primary key of the **DIAGNOSIS** entity, **Diagnosis_ID**, is added to the "**REQUIRES**" relation. Also, the primary key of the **MEDICAL_TEST** entity, **Test_ID**, is added. The combination of both keys represents the primary key of the "**REQUIRES**" relation.

5. CONTAINS1

Presc_ID	Doctor_ID	Patient_ID	Med_ID

Many prescriptions contain several medications. Thus "**CONTAINS1**" relationship is needed to link the **PRESCRIPTION** entity and the **MEDICATION** entity. In this relationship, the primary keys of the **PRESCRIPTION** and **MEDICATION** entities are included. The partial key of the **PRESCRIPTION** entity, **Presc_ID**, is added to the **"CONTAINS1"** relation. In addition, the primary key of the **DOCTOR** entity, **ID**, was renamed as **DOCTOR_ID** and added to this relation, as well as the primary key of the **PATIENT** entity, **ID**, renamed as **PATIENT_ID**. Also, the primary key of the **MEDICATION** entity, **Med_ID**, is added. The combination of all these keys represents the primary key of the "**CONTAINS1**" relation.

6. TREATS

Doctor_ID	Patient_ID

The "**TREATS**" relationship links the **DOCTOR** and **PATIENT** entities. A doctor can treat multiple patients, and a patient can be treated by multiple doctors. This is a many-to-many relationship. To represent this, a new relation called "**TREATS**" is created, which includes the primary keys **ID** renamed as **Doctor_ID** and another **ID** renamed as **Patient_ID** of the **DOCTOR** entity and **PATIENT** entity**,** respectively. The combination of both keys represents the primary key of the "**TREATS**" relation.

7. UNDERGOES

Patient_ID	Treatment_ID	Date

Many patients undergo many treatments. Thus, the “**UNDERGOES**” relationship is needed to link the **PATIENT** entity and the **TREATMENT** entity. In this relationship, the primary keys of the **PATIENT** and **TREATMENT** entities are included. The primary key of the **PATIENT** entity, **ID**, is renamed as **Patient_ID** and added to the “**UNDERGOES**” relation. Also, the primary key of the **TREATMENT** entity, **Treatment_ID**, is added. The “**UNDERGOES**” relationship has an attribute, **Date**, which is included in this relation. The combination of **Patient_ID** and **Treatment_ID** represents the primary key of the “**UNDERGOES**” relation.

STEP 6: Mapping of Multivalued Attributes

In this step, we are going to map the multivalued attributes which we ignored before. For each multivalued attribute, we create a new relation containing the related attribute and the primary key of the entity to which it belongs. Their combination will represent the primary key of the newly created relation. Our multivalued attributes are: **DOCTOR_PHONE_NUMBER**, **NURSE_PHONE_NUMBER**, and **PATIENT_PHONE_NUMBER**.

1. DOCTOR_PHONE_NUMBER

Doctor_ID	Phone_Number

The multivalued attribute **Phone_Number** belongs to the **DOCTOR** entity. To represent it, a "**DOCTOR_PHONE_NUMBER**" relation is created containing the related attribute and the primary key of the **DOCTOR** entity. The primary key of the **DOCTOR** entity is renamed **Doctor_ID** and added to this relation. The **Phone_Number** attribute is also added, which represents the multiple phone numbers that the doctor might have.

2. NURSE_PHONE_NUMBER

Nurse_ID	Phone_Number

The multivalued attribute **Phone_Number** belongs to the **NURSE** entity. To represent it, a "**NURSE_PHONE_NUMBER**" relation is created containing the related attribute and the primary key of the **NURSE** entity. The primary key of the **NURSE** entity is renamed **Nurse_ID** and added to this relation. The **Phone_Number** attribute is also added, which represents the multiple phone numbers that the nurse might have.

3. PATIENT_PHONE_NUMBER

Patient_ID	Phone_Number

The multivalued attribute **Phone_Number** belongs to the **PATIENT** entity. To represent it, a "**PATIENT_PHONE_NUMBER**" relation is created containing the related attribute and the primary key of the **PATIENT** entity. The primary key of the **PATIENT** entity is renamed **PATIENT_ID** and added to this relation. The **Phone_Number** attribute is also added, which represents the multiple phone numbers that the patient might have.

STEP 7: Mapping of N-ary Relationship Types

All N-ary relationship types are mapped in this step. A new relation containing the primary keys of all participating entities and any simple attributes of the relationship type is created. In our design, we have no N-ary relationship types, so this step is not applicable here.

Final Step: Final Displays

DOCTOR

ID	First_Name	Middle_Name	Last_Name	Email	Address	DOB	Gender	Specialization	Salary

NURSE

ID	First_Name	Middle_Name	Last_Name	Email	Address	Specialization	DOB	Gender	Shift	Salary

PATIENT

ID	First_Name	Middle_Name	Last_Name	Email	Address	DOB	Emergency_Contact	Gender

DEPARTMENT

Department_ID	Department_Name

APPOINTMENT

Appointment_Nb	App_Date	Time

TREATMENT

Treatment_ID	Type	Procedure	Requirements

DIAGNOSIS

Diagnosis_ID	Condition_Name	Description	Diagnosis_Date

MEDICAL_TEST

Test_ID	Name	Test_Result	Test_Date	Requirements

PHARMACY

Pharmacy_ID	Phone_Number	Pharmacy_Name	Address	Operating_Hours

ALLERGY

Allergy_ID	Allergen_Name	Description	Symptoms	Severity

MEDICAL_HISTORY

Medical_History_ID	Date	Details	Weight	Height	Blood_Type	Blood_Pressure	Blood_Glucose_Level

MEDICATION

Med_ID	Name	Description	Type	Dosage

ROOM

Room_Number	Room type	Capacity	Janitor_ID	Location

MEDICAL_EQUIPMENT

Serial_Number	Equipment_Name	Status	Storage_Conditions	Company_Provider

BILLING

Bill_ID	Bill_Date	Amount	Payment_Method	Payment_Status

PRESCRIPTION

Presc_ID	Doctor_ID	Patient_ID	Description	Presc_Date

MEDICAL_HISTORY (HAS_A1)

Medical_History_ID	Patient_ID	Date	Details	Height	Weight	Blood_Glucose_Level	Blood_Pressure	Blood_Type	Doctor_ID

BILLING (PAYS)

Bill_ID	Patient_ID	Bill_Date	Amount	Payment_Status	Payment_Method	App_ID

DOCTOR (BELONGS_TO1)

ID	Department_ID	First_Name	Middle_Name	Last_Name	Email	Address	DOB	Gender
Specialization	Salary							

ALLERGY (HAS_A2)

Allergy_ID	Patient_ID	Allergen_Name	Description	Symptoms	Severity

APPOINTMENT (HAS_A3)

Appointment_Nb	Patient_ID	Day	Month	Year	Hour	Minute	Doctor_ID

DIAGNOSIS (HAS_A4)

Diagnosis_ID	Patient_ID	Condition_Name	Description	Doctor_ID	Diagnosis_Date

PRESCRIPTION (ISSUES)

Presc_ID	Doctor_ID	Presc_Date	Patient_ID	Description	Issue_Date

MEDICAL_EQUIPMENT (CONTAINS2)

Serial_Number	Room_Number	Equipment_Name	Status	Company_Provider	Storage_Conditions

MEDICATION (SUPPLIES)

Med_ID	Pharmacy_ID	Name	Dosage	Type	Description

TREATMENT (PERFORMS)

Treatment_ID	Doctor_ID	Type	Procedure	Patient_ID	Requirements

TREATMENT (OCCURS_IN)

Treatment_ID	Room_Number	Type	Procedure	Patient_ID	Doctor_ID

PATIENT (RESIDES_IN)

ID	Room_Number	First_Name	Middle_Name	Last_Name	Email
Address	DOB	Gender	Emergency_Contact		

BELONGS_TO2

Nurse_ID	Department_ID	Start_Date	End_Date

SERVICES

Nurse_ID	Patient_ID

VISITS

Patient_ID	Doctor_ID	Visit_Date	Visit_Reason

REQUIRES

Test_ID	Diagnosis_ID

CONTAINS1

Presc_ID	Doctor_ID	Patient_ID	Med_ID

TREATS

Doctor_ID	Patient_ID

UNDERGOES

Patient_ID	Treatment_ID	Date

DOCTOR_PHONE_NUMBER

Doctor_ID	Phone_Number

NURSE_PHONE_NUMBER

Nurse_ID	Phone_Number

PATIENT_PHONE_NUMBER

Patient_ID	Phone_Number

X. Table Structure for The Clinic of Lost Souls:

After mapping our ER diagram into a relational database design, it is time to start creating the actual tables for our database on the Oracle Database Server. We will start by creating all tables and then inserting data into them. Finally, we will execute some queries to display the database's importance, especially in this clinic.

1. Doctor

```
CREATE TABLE DOCTOR
      (
          ID                    CHAR(9) PRIMARY KEY,
          FIRST_NAME            VARCHAR(15) NOT NULL,
          MIDDLE_NAME           VARCHAR(15),
          LAST_NAME             VARCHAR(15) NOT NULL,
          EMAIL                 VARCHAR(30),
          ADDRESS               VARCHAR(50),
          DOB                   DATE,
          GENDER                CHAR NOT NULL CHECK (GENDER IN ('F', 'M')),
          SPECIALIZATION        VARCHAR(100),
          SALARY                NUMBER(6,2),
          DEPARTMENT_ID         CHAR(9) NOT NULL
     );
```

2. Nurse

```
CREATE TABLE NURSE
      (
          ID                  CHAR(9) PRIMARY KEY,
          FIRST_NAME          VARCHAR(15) NOT NULL,
          MIDDLE_NAME         VARCHAR(15),
          LAST_NAME           VARCHAR(15) NOT NULL,
          EMAIL               VARCHAR(30),
          ADDRESS             VARCHAR(50),
          DOB                 DATE,
          GENDER              CHAR NOT NULL CHECK (GENDER IN ('F', 'M')),
          SPECIALIZATION      VARCHAR(100),
          SALARY              NUMBER(6,2),
          SHIFT               VARCHAR(15)
     );
```

3. Patient

```
CREATE TABLE PATIENT
      (
          ID                  CHAR(9) PRIMARY KEY,
          FIRST_NAME          VARCHAR(15) NOT NULL,
          MIDDLE_NAME         VARCHAR(15),
          LAST_NAME           VARCHAR(15) NOT NULL,
          EMAIL               VARCHAR(30),
          ADDRESS             VARCHAR(50),
          DOB                 DATE,
          GENDER              CHAR NOT NULL CHECK (GENDER IN ('F', 'M')),
          EMERGENCY_CONTACT   VARCHAR(50),
          ROOM_NUMBER         CHAR(9)
     );
```

4. Department

```
CREATE TABLE DEPARTMENT
      (
          DEPARTMENT_ID       CHAR(9) PRIMARY KEY,
          DEPARTMENT_NAME     VARCHAR(50) NOT NULL
     );
```

5. Appointment

```
CREATE TABLE APPOINTMENT
      (
          APPOINTMENT_NB   CHAR(9) PRIMARY KEY,
          APP_DATE         DATE NOT NULL,
          TIME             TIMESTAMP,
          PATIENT_ID       CHAR(9) NOT NULL,
          DOCTOR_ID        CHAR(9) NOT NULL,
          FOREIGN KEY      (PATIENT_ID) REFERENCES PATIENT(ID),
          FOREIGN KEY      (DOCTOR_ID) REFERENCES DOCTOR(ID)
     );
```

6. Treatment

```
 CREATE TABLE TREATMENT
      (
          TREATMENT_ID     CHAR(9) PRIMARY KEY,
          TYPE             VARCHAR(300),
          PROCEDURE        VARCHAR(300),
          PATIENT_ID       CHAR(9) NOT NULL,
          DOCTOR_ID        CHAR(9) NOT NULL,
          REQUIREMENTS     VARCHAR(300),
          ROOM_NUMBER      CHAR(9),
          FOREIGN KEY (PATIENT_ID) REFERENCES PATIENT(ID),
          FOREIGN KEY (DOCTOR_ID) REFERENCES DOCTOR(ID)
     );
```

7. Diagnosis

```
 CREATE TABLE DIAGNOSIS
       (
           DIAGNOSIS_ID      CHAR(9) PRIMARY KEY,
           CONDITION_NAME    VARCHAR(300),
           DESCRIPTION       VARCHAR(300),
           PATIENT_ID        CHAR(9) NOT NULL,
           DOCTOR_ID         CHAR(9) NOT NULL,
           DIAGNOSIS_DATE    DATE NOT NULL,
           FOREIGN KEY (PATIENT_ID) REFERENCES PATIENT(ID) ,
           FOREIGN KEY (DOCTOR_ID) REFERENCES DOCTOR(ID)
      );
```

8. Medical_Test

```
CREATE TABLE MEDICAL_TEST
       (
           TEST_ID        CHAR(9) PRIMARY KEY,
           NAME           VARCHAR(300) NOT NULL,
           TEST_RESULT    VARCHAR(300),
           PATIENT_ID     CHAR(9) NOT NULL,
           DOCTOR_ID      CHAR(9) NOT NULL,
           TEST_DATE      DATE NOT NULL,
           REQUIREMENTS   VARCHAR(300),
           FOREIGN KEY (PATIENT_ID) REFERENCES PATIENT(ID) ,
           FOREIGN KEY (DOCTOR_ID) REFERENCES DOCTOR(ID)
      );
```

9. Pharmacy

```
CREATE TABLE PHARMACY
      (
          PHARMACY_ID         CHAR(6) PRIMARY KEY,
          PHARMACY_NAME       VARCHAR(50) NOT NULL,
          PHONE_NUMBER        VARCHAR(15) NOT NULL,
          ADDRESS             VARCHAR(100),
          OPERATING_HOURS     VARCHAR(15)
     );
```

10. Allergy

```
CREATE TABLE ALLERGY
      (
          ALLERGY_ID        CHAR(9) PRIMARY KEY,
          ALLERGEN_NAME     VARCHAR(50) NOT NULL,
          DESCRIPTION       VARCHAR(50) NOT NULL,
          SEVERITY          VARCHAR(10) CHECK(Severity IN ('Low',
     'Moderate', 'High', 'Severe')),
          SYMPTOMS          VARCHAR(255),
          PATIENT_ID        CHAR(9)
     );
```

11. Medical_History

```
CREATE TABLE MEDICAL_HISTORY
      (
          MEDICAL_HISTORY_ID      CHAR(9) PRIMARY KEY,
          PATIENT_ID              CHAR(9) NOT NULL,
          MH_DATE                 DATE NOT NULL,
          DETAILS                 VARCHAR(255),
          DOCTOR_ID               CHAR(9) NOT NULL,
          WEIGHT                  NUMBER(5,2),
          HEIGHT                  NUMBER(5,2),
          BLOOD_TYPE              CHAR(3),
          BLOOD_PRESSURE          VARCHAR(10),
          BLOOD_GLUCOSE_LEVEL     NUMBER(5,2),
          FOREIGN KEY      (PATIENT_ID) REFERENCES PATIENT(ID),
          FOREIGN KEY      (DOCTOR_ID) REFERENCES DOCTOR(ID)
           );
```

12. Medication

```
CREATE TABLE MEDICATION
    (
          MED_ID              CHAR(9) PRIMARY KEY,
          NAME                VARCHAR(100) NOT NULL,
          DESCRIPTION         VARCHAR(255),
          TYPE                VARCHAR(50),
          DOSAGE              VARCHAR(50) NOT NULL,
          PHARMACY_ID         CHAR(10)
    );
```

13. Room

```
CREATE TABLE ROOM
      (
               ROOM_NUMBER     CHAR(9) PRIMARY KEY,
               ROOM_TYPE       VARCHAR (50) NOT NULL,
               CAPACITY        NUMBER NOT NULL,
               JANITOR_ID      CHAR(9),
               LOCATION        VARCHAR(100)
        );
```

14. Medical_Equipment

```
CREATE TABLE MEDICAL EQUIPMENT
    (
          SERIAL_NB                CHAR(9) PRIMARY KEY,
          EQUIPMENT_NAME           VARCHAR(50) NOT NULL,
          STATUS                   VARCHAR(20),
          STORAGE_CONDITIONS       VARCHAR(100),
          COMPANY_PROVIDER         VARCHAR(50),
          ROOM_NUMBER              CHAR(9) NOT NULL
    );
```

15. Billing

```
CREATE TABLE BILLING
 (
       BILL_ID          CHAR(9) PRIMARY KEY,
       BILL_DATE        DATE  NOT NULL,
       AMOUNT           NUMBER(10, 2) NOT NULL,
       PAYMENT_METHOD   VARCHAR(50),
       PATIENT_ID       CHAR(9) NOT NULL,
       APP_ID           CHAR(9),
       FOREIGN KEY      (PATIENT_ID) REFERENCES PATIENT(ID),
       FOREIGN KEY      (APP_ID) REFERENCES APPOINTMENT(APPOINTMENT_NB)
   );
```

16. Prescription

```
CREATE TABLE PRESCRIPTION
     (
         PRESC_ID         CHAR(6) NOT NULL,
         DESCRIPTION      VARCHAR(200),
         PATIENT_ID       CHAR(9) NOT NULL,
         DOCTOR_ID        CHAR(9) NOT NULL,
         PRESC_DATE       DATE NOT NULL,
         PRIMARY KEY     (DOCTOR_ID, PATIENT_ID, PRESC_ID),
         FOREIGN KEY     (DOCTOR_ID) REFERENCES DOCTOR(ID),
         FOREIGN KEY     (PATIENT_ID)  REFERENCES PATIENT(ID)
    );
```

17. Belongs_To2

```
CREATE TABLE BELONGS_TO2
      (
          NURSE_ID         CHAR(9) NOT NULL,
          DEPARTMENT_ID    CHAR(9) NOT NULL,
          START_DATE       DATE NOT NULL,
          END_DATE         DATE,
          PRIMARY KEY      (ID, Department_ID),
          FOREIGN KEY      (Nurse_ID) REFERENCES NURSE (ID),
          FOREIGN KEY      (Department_ID) REFERENCES DEPARTMENT
     (Department_ID)
     );
```

18. Services

```
CREATE TABLE SERVICES
      (
          NURSE_ID         CHAR(9) NOT NULL,
          PATIENT_ID       CHAR(9) NOT NULL,
          PRIMARY KEY      (Nurse_ID, Patient_ID),
          FOREIGN KEY      (Nurse_ID) REFERENCES NURSE (ID),
          FOREIGN KEY      (Patient_ID) REFERENCES PATIENT (ID)
     );
```

19. Visits

```
CREATE TABLE VISITS
      (
          PATIENT_ID      CHAR(9) NOT NULL,
          DOCTOR_ID       CHAR(9) NOT NULL,
          VISIT_DATE      DATE NOT NULL,
          VISIT_REASON    VARCHAR(255),
          PRIMARY KEY     (PATIENT_ID,DOCTOR_ID),
          FOREIGN KEY     (PATIENT_ID) REFERENCES PATIENT (ID),
          FOREIGN KEY     (DOCTOR_ID) REFERENCES DOCTOR (ID)
     );
```

20. Requires

```
CREATE TABLE REQUIRES
      (
          TEST_ID        CHAR(9) NOT NULL,
          DIAGNOSIS_ID   CHAR(9) NOT NULL,
          PRIMARY KEY    (TEST_ID, DIAGNOSIS_ID),
          FOREIGN KEY    (TEST_ID) REFERENCES MEDICAL_TEST (TEST_ID),
          FOREIGN KEY    (DIAGNOSIS_ID) REFERENCES DIAGNOSIS
(DIAGNOSIS_ID)
     );
```

21. Contains1

```
CREATE TABLE CONTAINS1
      (
          PRESC_ID       CHAR(10) NOT NULL,
          MED_ID         CHAR(9)  NOT NULL,
          PRIMARY KEY    (PRESC_ID, MED_ID)  ,
          FOREIGN KEY    (PRESC_ID) REFERENCES PRESCRIPTION (PRESC_ID),
          FOREIGN KEY    (MED_ID) REFERENCES MEDICATION(MED_ID)
      );
```

22. Treats

```
CREATE TABLE TREATS
    (
        DOCTOR_ID              CHAR(9) NOT NULL,
        PATIENT_ID             CHAR(9) NOT NULL,
        PRIMARY KEY           (DOCTOR_ID, PATIENT_ID),
        FOREIGN KEY           (DOCTOR_ID) REFERENCES DOCTOR (ID),
        FOREIGN KEY           (PATIENT_ID)  REFERENCES PATIENT (ID)
    );
```

23. Undergoes

```
CREATE TABLE UNDERGOES
    (
        PATIENT_ID         CHAR(9) NOT NULL,
        TREATMENT_ID       CHAR(9) NOT NULL,
        TREATMENT_DATE     DATE NOT NULL,
        PRIMARY KEY        (PATIENT_ID, TREATMENT_ID),
        FOREIGN KEY        (PATIENT _ID) REFERENCES PATIENT (ID),
        FOREIGN KEY        (TREATMENT_ID)  REFERENCES TREATMENT
(TREATMENT_ID)
    );
```

24. Doctor_Phone_Number

```
CREATE TABLE DOCTOR_PHONE_NUMBER
     (
         DOCTOR_ID             CHAR(9) NOT NULL,
         PHONE_NUMBER          VARCHAR(15) NOT NULL,
         PRIMARY KEY           (DOCTOR_ID, PHONE_NUMBER),
         FOREIGN KEY           (DOCTOR_ID) REFERENCES DOCTOR (ID)
     );
```

25. Nurse_Phone_Number

```
CREATE TABLE NURSE_PHONE_NUMBER
     (
          NURSE_ID                CHAR(9) NOT NULL,
          PHONE_NUMBER            VARCHAR(15) NOT NULL,
          PRIMARY KEY             (NURSE_ID, PHONE_NUMBER),
          FOREIGN KEY             (NURSE_ID) REFERENCES NURSE (ID)
     );
```

26. Patient_Phone_Number

```
CREATE TABLE PATIENT_PHONE_NUMBER
     (
          PATIENT_ID              CHAR(9) NOT NULL,
          PHONE_NUMBER            VARCHAR(15) NOT NULL,
          PRIMARY KEY             (PATIENT_ID, PHONE_NUMBER),
          FOREIGN KEY             (PATIENT_ID) REFERENCES PATIENT (ID)
     );
```

Some referential integrity constraints cannot be added directly upon creation of the table because the primary key it references is still not created because of that we are obliged to use the ALTER command.

```
ALTER TABLE DOCTOR
ADD CONSTRAINT FK_DEPARTMENT
FOREIGN KEY (DEPARTMENT_ID) REFERENCES DEPARTMENT(DEPARTMENT_ID);

ALTER TABLE PATIENT
ADD CONSTRAINT FK_ROOM
FOREIGN KEY (ROOM_NUMBER) REFERENCES ROOM(ROOM_NUMBER);

ALTER TABLE ALLERGY
ADD CONSTRAINT FK_PATIENT
FOREIGN KEY (PATIENT_ID) REFERENCES PATIENT(ID);

ALTER TABLE MEDICAL_EQUIPMENT
ADD CONSTRAINT FK_ROOM
FOREIGN KEY (ROOM_NUMBER) REFERENCES ROOM(ROOM_NUMBER);

ALTER TABLE TREATMENT
ADD CONSTRAINT FK_ROOM
FOREIGN KEY (ROOM_NUMBER) REFERENCES ROOM(ROOM_NUMBER);

ALTER TABLE MEDICATION
ADD CONSTRAINT FK_PHARMACY
FOREIGN KEY (PHARMACY_ID) REFERENCES PHARMACY(PHARMACY_ID);
```

XI. Table Descriptions:

After creating all the tables on the oracle database server, we can view the description of each table in order to make sure everything is fine and no mistakes were made during creation of table.

In our database, we have the following tables created on the oracle database server:

```
SQL> SELECT DISTINCT OBJECT_NAME
FROM USER_OBJECTS
WHERE OBJECT_TYPE = 'TABLE';

OBJECT_NAME

--------------------------------------------------------------------------------

SERVICES
NURSE
DOCTOR
MEDICATION
DIAGNOSIS
PATIENT
APPOINTMENT
PHARMACY
REQUIRES
VISITS
TREATMENT
PRESCRIPTION
BILLING
DOCTOR_PHONE_NUMBER
MEDICAL_EQUIPMENT
CONTAINS1
MEDICAL_TEST
TREATS
PATIENT_PHONE_NUMBER
DEPARTMENT
MEDICAL_HISTORY
BELONGS_TO2
ALLERGY
UNDERGOES
NURSE_PHONE_NUMBER
ROOM
```

1. Doctor

```
SQL> DESC DOCTOR;

Name                                Null?     Type

----------------------------------- --------- --------------

ID                                  NOT NULL  CHAR(9)

First_Name                          NOT NULL  VARCHAR2(15)

Middle_Name                         NULL      VARCHAR2(15)

Last_Name                           NOT NULL  VARCHAR2(15)

Email                               NULL      VARCHAR2(30)

Address                             NULL      VARCHAR(50)

DOB                                 NULL      DATE

Gender                              NOT NULL  CHAR(1)

Specialization                      NULL      VARCHAR(100)

Salary                              NULL      NUMBER(6,2)

Department_ID                       NOT NULL  CHAR(9)
```

2. Nurse

```
SQL> DESC NURSE;

Name                                      Null?       Type
---------------------------------------- ---------- -----------------
ID                                        NOT NULL      CHAR(9)
First_Name                                NOT NULL      VARCHAR2(15)
Middle_Name                               NULL          VARCHAR2(15)
Last_Name                                 NOT NULL      VARCHAR2(15)
Email                                     NULL          VARCHAR2(30)
Address                                   NULL          VARCHAR2(50)
DOB                                       NULL          DATE
Gender                                    NOT NULL      CHAR(1)
Specialization                            NULL          VARCHAR2(100)
Salary                                    NULL          NUMBER(6,2)
Shift                                     NULL          VARCHAR2(15)
```

3. Patient

```
SQL> DESC PATIENT;

Name                                      Null?     Type
----------------------------------------- --------- -----------------
ID                                        NOT NULL  CHAR(9)
First_Name                                NOT NULL  VARCHAR2(15)
Middle_Name                               NULL      VARCHAR2(15)
Last_Name                                 NOT NULL  VARCHAR2(15)
Email                                     NULL      VARCHAR2(30)
Address                                   NULL      VARCHAR2(50)
DOB                                       NULL      DATE
Gender                                    NOT NULL  CHAR(1)
Emergency_Contact                         NULL      VARCHAR2(50)
Room_Number                               NULL      CHAR(9)
```

4. Department

```
SQL> DESC DEPARTMENT;

Name                                      Null?     Type
----------------------------------------- --------- -----------------
Department_ID                              NOT NULL    CHAR(9)
Department_Name                            NOT NULL    VARCHAR2(50)
```

5. Appointment

```
SQL> DESC APPOINTMENT;


Name                               Null?     Type

---------------------------------- --------  --------------

Appointment_Nb                     NOT NULL  CHAR(9)

App_Date                           NOT NULL  DATE

Time                               NULL      TIMESTAMP

Patient_ID                         NOT NULL  CHAR(9)

Doctor_ID                          NOT NULL  CHAR(9)
```

6. Treatment

```
SQL> DESC TREATMENT;


Name                               Null?     Type

---------------------------------- --------  --------------

Treatment_ID                       NOT NULL  CHAR(9)

Type                               NULL      VARCHAR2(300)

Procedure                          NULL      VARCHAR2(300)

Patient_ID                         NOT NULL  CHAR(9)

Doctor_ID                          NOT NULL  CHAR(9)

Requirements                       NULL      VARCHAR2(300)

Room_Number                        NULL      CHAR(9)
```

7. Diagnosis

```
SQL> DESC DIAGNOSIS;

Name                                      Null?      Type
----------------------------------------- ---------- -----------------
Diagnosis_ID                              NOT NULL   CHAR(9)
Condition_Name                            NULL       VARCHAR2(300)
Description                               NULL       VARCHAR2(300)
Patient_ID                                NOT NULL   CHAR(9)
Doctor_ID                                 NOT NULL   CHAR(9)
Diagnosis_Date                            NOT NULL   DATE
```

8. Medical_Test

```
SQL> DESC MEDICAL_TEST;

Name                                      Null?      Type
----------------------------------------- ---------- -----------------
Test_ID                                   NOT NULL   CHAR(9)
Name                                      NOT NULL   VARCHAR2(300)
Test_Result                               NULL       VARCHAR2(300)
Patient_ID                                NOT NULL   CHAR(9)
Doctor_ID                                 NOT NULL   CHAR(9)
Test_Date                                 NOT NULL   DATE
Requirements                              NULL       VARCHAR2(300)
```

9. Pharmacy

```
SQL> DESC  PHARMACY;

Name                                       Null?     Type

----------------------------------------- ---------- -----------------

Pharmacy_ID                                NOT NULL   CHAR(6)

Pharmacy_Name                              NOT NULL   VARCHAR2(50)

Phone_Number                               NOT NULL   VARCHAR2(15)

Address                                    NULL       VARCHAR2(100)

Operating_Hours                            NULL       VARCHAR2(15)
```

10. Allergy

```
SQL> DESC  ALLERGY;

Name                                       Null?     Type

----------------------------------------- ---------- -----------------

Allergy_ID                                 NOT NULL   CHAR(9)

Allergen_Name                              NOT NULL   VARCHAR2(50)

Description                                NOT NULL   VARCHAR2(50)

Severity                                   NULL       VARCHAR2(10)

Symptoms                                   NULL       VARCHAR2(255)

Patient_ID                                 NOT NULL  CHAR(9)
```

11. Medical_History

```
SQL> DESC MEDICAL_HISTORY;

Name                                   Null?     Type
-------------------------------------- --------- ---------------
Medical_History_ID                      NOT NULL  CHAR(9)
Patient_ID                              NOT NULL  CHAR(9)
MH_Date                                 NOT NULL  DATE
Details                                 NULL      VARCHAR2(255)
Doctor_ID                               NOT NULL  CHAR(9)
Weight                                  NULL      NUMBER(5,2)
Height                                  NULL      NUMBER(5,2)
Blood_Type                              NULL      CHAR(3)
Blood_Pressure                          NULL      VARCHAR2(10)
Blood_Glucose_Level                     NULL      NUMBER(5,2)
```

12. Medication

```
SQL> DESC MEDICATION;

Name                                   Null?     Type
-------------------------------------- --------- ---------------
Med_ID                                 NOT NULL  CHAR(9)
Name                                   NOT NULL  VARCHAR2 (100)
Description                            NULL      VARCHAR2 (255)
Type                                   NULL      VARCHAR2 (50)
Dosage                                 NOT NULL  VARCHAR2 (50)
Pharmacy_ID                            NOT NULL  CHAR(6)
```

13. Room

```
SQL> DESC ROOM;


Name                                   Null?     Type

-------------------------------------- --------- ----------------

Room_Number                            NOT NULL  CHAR(9)
Room_Type                              NOT NULL  VARCHAR2(50)
Capacity                               NOT NULL  NUMBER
Janitor_ID                             NULL      CHAR(9)
Location                               NULL      VARCHAR2(100)
```

14. Medical_Equipment

```
SQL> DESC MEDICAL_EQUIPMENT;


Name                                   Null?     Type

-------------------------------------- --------- ----------------

Serial_Nb                               NOT NULL  CHAR(9)
Equipment_Name                          NOT NULL  VARCHAR2(50)
Status                                  NULL      VARCHAR2(20)
Storage_Conditions                      NULL      VARCHAR2(100)
Company_Provider                        NULL      VARCHAR2(50)
Room_Number                             NOT NULL  CHAR(9)
```

15. Billing

```
SQL> DESC BILLING;

Name                                      Null?      Type

----------------------------------------- ---------- -----------------

Bill_ID                                   NOT NULL   CHAR(9)
Bill_Date                                 NOT NULL   DATE
Amount                                    NOT NULL   NUMBER(10,2)
Payment_Method                            NULL       VARCHAR2(50)
Patient_ID                                NOT NULL   CHAR(9)
App_ID                                    NULL       CHAR(9)
```

16. Prescription

```
SQL> DESC PRESCRIPTION;

Name                                       Null?     Type

----------------------------------------- ---------- -----------------

Presc_ID                                  NOT NULL   CHAR(6)

Description                               NULL       VARCHAR2(200)

Patient_ID                                NOT NULL   CHAR(9)

Doctor_ID                                 NOT NULL   CHAR(9)

Presc_Date                                NOT NULL   DATE
```

17. Belongs_To2

```
SQL> DESC BELONGS_TO2;

Name                                      Null?      Type

----------------------------------------- ---------- -----------------

ID                                        NOT NULL   CHAR(9)
Department_ID                             NOT NULL   CHAR(9)
Start_Date                                NOT NULL   DATE
End_Date                                  NULL       DATE
```

18. Services

```
SQL> DESC SERVICES;

Name                                      Null?      Type

----------------------------------------- ---------- -----------------

Nurse_ID                                  NOT NULL   CHAR(9)
Patient_ID                                NOT NULL   CHAR(9)
```

19. Visits

```
SQL> DESC VISITS;

Name                                      Null?      Type

----------------------------------------- ---------- -----------------

Patient_ID                                NOT NULL   CHAR(9)
Doctor_ID                                 NOT NULL   CHAR(9)
```

```
Visit_Date                                NOT NULL   DATE
Visit_Reason                              NULL       VARCHAR2 (255)
```

20. Requires

```
SQL> DESC REQUIRS;


Name                                     Null?      Type

---------------------------------------- ---------- ----------------

Test_ID                                  NOT NULL   CHAR(9)
Diagnosis_ID                             NOT NULL   CHAR(9)
```

21. Contains1

```
SQL> DESC CONTAINS1;


Name                                     Null?      Type

---------------------------------------- ---------- ----------------

Presc_ID                                 NOT NULL   CHAR(10)
Med_ID                                   NOT NULL   CHAR(9)
```

22. Treats

```
SQL> DESC TREATS;

Name                                     Null?      Type


---------------------------------------- ---------- -------------
Patient_ID                                NOT NULL   CHAR(9)
Doctor_ID                                 NOT NULL   CHAR(9)
```

23. Undergoes

```
SQL> DESC  UNDERGOES;

Name                                      Null?     Type
----------------------------------------- --------- -----------------
Patient_ID                                 NOT NULL  CHAR(9)
Treatment_ID                               NOT NULL  CHAR(9)
Treatment_Date                             NOT NULL  DATE
```

24. Doctor_Phone_Number

```
SQL> DESC  DOCTOR_PHONE_NUMBER;

Name                                      Null?     Type
----------------------------------------- --------- -----------------
Doctor_ID                                 NOT NULL  CHAR(9)
Phone_Number                              NOT NULL  VARCHAR2(15)
```

25. Nurse_Phone_Number

```
SQL> DESC  NURSE_PHONE_NUMBER;

Name                                      Null?     Type
----------------------------------------- --------- -----------------
Nurse_ID                                  NOT NULL  CHAR(9)
Phone_Number                              NOT NULL  VARCHAR2(15)
```

26. Patient_Phone_Number

```
SQL> DESC PATIENT_PHONE_NUMBER;


Name                                   Null?     Type

-------------------------------------- --------- ---------------

Patient_ID                             NOT NULL   CHAR(9)
Phone_Number                           NOT NULL  VARCHAR2(15)
```

XII. Inserting Data:

1. Doctor

```
INSERT INTO DOCTOR VALUES('DR-000001', 'Hassan', 'Mohammad', 'Itani',
'hassan.itani@gmail.com', 'Beirut', TO_DATE ('06-23-1994', 'MM-DD-YYYY'), 'M',
'Cardiology', 7000.00, 'DP-000001');

INSERT INTO DOCTOR VALUES('DR-000002', 'John', 'David', 'Smith', 'john.smith@gmail.com',
'Saida', TO_DATE ('09-05-1982', 'MM-DD-YYYY'), 'M', 'Digestive System', 7500.00, 'DP-
000002');

INSERT INTO DOCTOR VALUES('DR-000003', 'Sally', 'Ibrahim', 'Hamade',
'sally.hamade@gmail.com', 'Tyre', TO_DATE ('08-31-1990', 'MM-DD-YYYY'), 'F', 'Internal
Medicine', 6000.00, 'DP-000003');

INSERT INTO DOCTOR VALUES('DR-000004', 'Wael', 'Fouad', 'Khoury',
'wael.khoury@gmail.com', 'Shouf', TO_DATE ('06-02-1992', 'MM-DD-YYYY'), 'M',
'Immunology', 9000.00, 'DP-000004');

INSERT INTO DOCTOR VALUES('DR-000005', 'Maha', 'Ahmad', 'Salem', 'maha.salem@gmail.com',
'Beirut', TO_DATE ('03-01-1975', 'MM-DD-YYYY'), 'F', 'Pediatrics', 7750.00, 'DP-000005');

INSERT INTO DOCTOR VALUES('DR-000006', 'Mohammad', 'Bassam', 'Ismail',
'mohammad.ismail@gmail.com', 'Jbeil', TO_DATE ('12-08-1968', 'MM-DD-YYYY'), 'M',
'Ophthalmology', 6500.00, 'DP-000006');

INSERT INTO DOCTOR VALUES('DR-000007', 'Hiba', 'Sami', 'Rashidi',
'hiba.rashidi@gmail.com', 'Tripoli', TO_DATE ('02-20-1977', 'MM-DD-YYYY'), 'F',
'Surgery', 9860.00, 'DP-000007');

INSERT INTO DOCTOR VALUES('DR-000008', 'Hisham', 'Youssef', 'Al Majzoub',
'hisham.youssef@gmail.com', 'Bekaa', TO_DATE ('06-04-1980', 'MM-DD-YYYY'), 'M',
'Radiology', 8900.00, 'DP-000008');

INSERT INTO DOCTOR VALUES('DR-000009', 'Nour', 'Khaled', 'Ramadan',
'nour.ramadan@gmail.com', 'Beirut', TO_DATE ('07-05-1969', 'MM-DD-YYYY'), 'F',
'Obstetrics', 7850.00, 'DP-000009');

INSERT INTO DOCTOR VALUES('DR-000010', 'Karim', 'Malek', 'Ghandour',
'karim.ghandour@gmail.com', 'Bhamdoun', TO_DATE ('11-08-1972', 'MM-DD-YYYY'), 'M',
'Psychology', 6650.00, 'DP-000010');

INSERT INTO DOCTOR VALUES('DR-000011', 'Asmaa', 'Walid', 'Mehio',
'asmaa.mehio@gmail.com', 'Tyre', TO_DATE ('06-12-1990', 'MM-DD-YYYY'), 'F', 'Internal
Medicine', 6000.00, 'DP-000003');
```

2. Nurse

```
INSERT INTO NURSE VALUES('NS-000001', 'Reem', 'Jihad', 'Sidani', 'reem.sidani@gmail.com',
'Beirut', TO_ DATE ('02-07-1999', 'MM-DD-YYYY'), 'F', 'Cardiac Care Nursing, Certified
Heart Failure Nurse', 4500.00, '8 AM – 6 PM');

INSERT INTO NURSE VALUES('NS-000002', 'Adam', 'Issa', 'Al Rifai',
'adam.alrifai@gmail.com', 'Saida', TO_DATE ('10-11-1973', 'MM-DD-YYYY'), 'M',
'Gastroenterology Nursing', 5570.00,  '9 AM – 5 PM');

INSERT INTO NURSE VALUES('NS-000003', 'Reina', 'Kamel', 'Shaaban',
'reina.shaaban@gmail.com', 'Tyre', TO_DATE ('04-09-1990', 'MM-DD-YYYY'), 'F', 'Certified
Nurse, Assessment and Monitoring Skills, Communication Skills', 6000.00, '9 AM – 6 PM');

INSERT INTO NURSE VALUES('NS-000004', 'Klara', 'Bassam', 'Ghannoum',
'klara.ghannoum@gmail.com', 'Beirut', TO_DATE ('06-25-2000', 'MM-DD-YYYY'), 'F',
'Certified Immunology Nurse, ', 6250.00, '12 PM – 11 PM');

INSERT INTO NURSE VALUES('NS-000005', 'Tarek', 'Imad', 'Mousa', 'tarek.mousa@gmail.com',
'Beirut', TO_DATE ('09-05-1987', 'MM-DD-YYYY'), 'M', 'Certified Radiology Nurse, MRI
Nurse', 6200.00, '8 AM – 5 PM');

INSERT INTO NURSE VALUES('NS-000006', 'Samira', 'Wafiq', 'Rashad',
'samira.rashad@gmail.com', 'Jbeil', TO_DATE ('03-08-1990', 'MM-DD-YYYY'), 'F', 'Certified
Surgical Nurse, Post-operative Care', 6500.00, '9 AM – 6 PM');

INSERT INTO NURSE VALUES('NS-000007', 'Teya', 'Karam', 'Sinno', 'teya.sinno@gmail.com',
'Tripoli', TO_DATE ('05-24-1993', 'MM-DD-YYYY'), 'F', 'Certified Nurse in Obstetrics,
Prenatal Care Skills, Labor and Delivery Support', 6640.00, '11 AM – 6 PM');

INSERT INTO NURSE VALUES('NS-000008', 'Nabil', 'Omar', 'Yamout',
'nabil.yamout@gmail.com', 'Bekaa', TO_DATE ('06-09-1980', 'MM-DD-YYYY'), 'M', 'Certified
Mental Health Nurse, Great Attentive and Monitoring Skills', 6000.00, '9 AM – 3 PM');

INSERT INTO NURSE VALUES('NS-000009', 'Amani', 'Ziad', 'Itani', 'amani.itani@gmail.com',
'Beirut', TO_DATE ('10-27-1969', 'MM-DD-YYYY'), 'F', 'Certified Surgical Assistant,
Handling Surgical Instruments, Attentive and Teamwork Skills', 6700.00, '8 AM – 7 PM');

INSERT INTO NURSE VALUES('NS-000010', 'Sara', 'Raed', 'Monter', 'sara.monter@gmail.com',
'Bhamdoun', TO_DATE ('04-02-1996', 'MM-DD-YYYY'), 'F', 'Assessment and Monitoring Skills,
Children Care Skills', 5750.00, '9 AM – 5 PM');
```

3. Patient

```
INSERT INTO PATIENT VALUES('PT-000001', 'Lana', 'Moustafa', 'Al Saleh',
'lana.alsaleh@gmail.com', 'Beirut', TO_DATE ('04-05-1986', 'MM-DD-YYYY'), 'F', 'Moustafa
Al Saleh: 71-923331', 'RM-000001');

INSERT INTO PATIENT VALUES('PT-000002', 'Issam', 'Hamed', 'Badawi',
'issam.badawi@gmail.com', 'Jbeil', TO_DATE ('08-28-1986', 'MM-DD-YYYY'), 'M', 'Nour
Badawi: 03-167588', 'RM-000002');

INSERT INTO PATIENT VALUES('PT-000003', 'Tamara', 'Hani', 'Hamdan',
'tamara.hamdan@gmail.com', 'Saida', TO_DATE ('05-31-1986', 'MM-DD-YYYY'), 'F', 'Aya
Hamdan: 71-876673', 'RM-000002');

INSERT INTO PATIENT VALUES('PT-000004', 'Jad', 'Wassim', 'Halabi',
'jad.halabi@gmail.com', 'Tyre', TO_DATE ('04-29-1995', 'MM-DD-YYYY'), 'M', 'Wassim
Halabi: 71-996339', NULL);

INSERT INTO PATIENT VALUES('PT-000005', 'Rea', 'Amer', 'Bisre', 'rea.bisre@gmail.com',
'Tripoli', TO_DATE ('01-26-1986', 'MM-DD-YYYY'), 'F', 'Lea Bisre: 03-442366', 'RM-
000009');

INSERT INTO PATIENT VALUES('PT-000006', 'Khaled', 'Hamza', 'Daaboul',
'khaled.daaboul@gmail.com', 'Jbeil', TO_DATE ('12-09-2006', 'MM-DD-YYYY'), 'M', 'Hamza
Daaboul: 76-128865', NULL);

INSERT INTO PATIENT VALUES('PT-000007', 'Rawan', 'Oussama', 'Tabesh',
'rawan.tabesh@gmail.com', 'Tripoli', TO_ DATE ('11-20-1986', 'MM-DD-YYYY'), 'F', 'Oussama
Arnaout: 71-445511', 'RM-000008');

INSERT INTO PATIENT VALUES('PT-000008', 'Zaid', 'Ali', 'El Taweel',
'zaid.eltaweel@gmail.com', 'Bhamdoun', TO_ DATE ('02-08-1946', 'MM-DD-YYYY'), 'M', 'Ali
El Taweel: 71-60123', 'RM-000004');

INSERT INTO PATIENT VALUES('PT-000009', 'Katia', 'Bassam', 'Ghannoum',
'katia.ghannoum@gmail.com', 'Beirut', TO_ DATE ('04-29-1999', 'MM-DD-YYYY'), 'F', 'Klara
Ghannoum: 76-178965', 'RM-000007');

INSERT INTO PATIENT VALUES('PT-000010', 'John', 'James', 'Doe', 'john.doe@gmail.com',
'Beirut', TO_ DATE ('07-31-1986', 'MM-DD-YYYY'), 'M', 'Jane Doe: 03-315415', 'RM-
000003');
```

4. Department

```
INSERT INTO DEPARTMENT VALUES ('DP-000001', 'Cardiology);

INSERT INTO DEPARTMENT VALUES ('DP-000002', 'Digestive System');

INSERT INTO DEPARTMENT VALUES('DP-000003', 'Internal Medicine');

INSERT INTO DEPARTMENT VALUES('DP-000004', 'Immunology');

INSERT INTO DEPARTMENT VALUES('DP-000005', 'Pediatrics');

INSERT INTO DEPARTMENT VALUES('DP-000006', 'Ophthalmology');

INSERT INTO DEPARTMENT VALUES('DP-000007', 'Surgery');

INSERT INTO DEPARTMENT VALUES('DP-000008', 'Radiology');

INSERT INTO DEPARTMENT VALUES('DP-000009', 'Obstetrics');

INSERT INTO DEPARTMENT VALUES('DP-000010', 'Psychology');
```

5. Appointment

```
INSERT INTO APPOINTMENT VALUES ('AP-000001', TO_DATE ('11-01-2024', 'MM-DD-YYYY'),
TIMESTAMP'2024-11-01 14:00:00', 'PT-000001', 'DR-000001');

INSERT INTO APPOINTMENT VALUES('AP-000002', TO_DATE ('11-02-2024', 'MM-DD-YYYY'),
TIMESTAMP'2024-11-02 12:00:00', 'PT-000002', 'DR-000002');

INSERT INTO APPOINTMENT VALUES('AP-000003', TO_DATE ('11-03-2024', 'MM-DD-YYYY'),
TIMESTAMP'2024-11-03 13:00:00', 'PT-000003', 'DR-000003');

INSERT INTO APPOINTMENT VALUES('AP-000004', TO_DATE ('11-04-2024', 'MM-DD-YYYY') ,
TIMESTAMP'2024-11-04 14:00:00', 'PT-000004', 'DR-000004');

INSERT INTO APPOINTMENT VALUES('AP-000005', TO_DATE ('11-05-2024', 'MM-DD-YYYY'),
TIMESTAMP'2024-11-05 16:00:00', 'PT-000005', 'DR-000005');

INSERT INTO APPOINTMENT VALUES('AP-000006', TO_DATE ('11-06-2024', 'MM-DD-YYYY'),
TIMESTAMP'2024-11-06 08:00:00', 'PT-000006', 'DR-000006');

INSERT INTO APPOINTMENT VALUES('AP-000007', TO_DATE ('11-07-2024', 'MM-DD-YYYY'),
TIMESTAMP'2024-11-07 14:00:00', 'PT-000007', 'DR-000007');

INSERT INTO APPOINTMENT VALUES('AP-000008', TO_DATE ('11-08-2024', 'MM-DD-YYYY'),
TIMESTAMP'2024-11-08 07:00:00', 'PT-000008', 'DR-000008');

INSERT INTO APPOINTMENT VALUES('AP-000009', TO_DATE ('11-09-2024', 'MM-DD-YYYY'),
TIMESTAMP'2024-11-09 11:00:00', 'PT-000009', 'DR-000009');

INSERT INTO APPOINTMENT VALUES('AP-000010', TO_DATE ('11-10-2024', 'MM-DD-YYYY'),
TIMESTAMP'2024-11-10 09:00:00', 'PT-000010', 'DR-000010');
```

```
INSERT INTO APPOINTMENT VALUES ('AP-000011', TO_DATE('04-05-2024', 'MM-DD-YYYY'),
TIMESTAMP '2024-04-05 10:00:00', 'PT-000001', 'DR-000001');

INSERT INTO APPOINTMENT VALUES ('AP-000012', TO_DATE ('11-01-2024', 'MM-DD-YYYY'),
TIMESTAMP'2024-11-01 16:00:00', 'PT-000003', 'DR-000001');
```

6. Treatment

```
INSERT INTO TREATMENT VALUES('TR-000001', 'Physical Therapy', 'Spinal Alignment', 'PT-
000001', 'DR-000001', 'Specialized Equipment Required', 'RM-000001');

INSERT INTO TREATMENT VALUES('TR-000002', 'Cardiology', 'Echocardiogram', 'PT-000002',
'DR-000002', 'Ultrasound Machine Availability', 'RM-000002');

INSERT INTO TREATMENT VALUES('TR-000003', 'Dermatology', 'Laser Treatment', 'PT-000003',
'DR-000003', 'Protective Eyewear Required', 'RM-000003');

INSERT INTO TREATMENT VALUES('TR-000004', 'Cardiology', 'Cardiac Arrest Resuscitation',
'PT-000004', 'DR-000004', 'Defibrillator Required', 'RM-000004');

INSERT INTO TREATMENT VALUES('TR-000005', 'Neurology', 'EEG Analysis', 'PT-000005', 'DR-
000005', 'Electrode Gel and Sensor Pads', 'RM-000005');

INSERT INTO TREATMENT VALUES('TR-000006', 'Gastroenterology', 'Colonoscopy', 'PT-000006',
'DR-000006', 'Pre-Treatment Diet Plan', 'RM-000006');

INSERT INTO TREATMENT VALUES('TR-000007', 'Oncology', 'Chemotherapy Session', 'PT-
000007', 'DR-000007', 'IV Setup and Medications', 'RM-000007');

INSERT INTO TREATMENT VALUES('TR-000008', 'Pediatrics', 'Vaccination', 'PT-000008', 'DR-
000008', 'Vaccine Storage and Needles', 'RM-000008');

INSERT INTO TREATMENT VALUES('TR-000009', 'Ophthalmology', 'Cataract Evaluation', 'PT-
000009', 'DR-000009', 'Eye Drops for Dilation', 'RM-000009');

INSERT INTO TREATMENT VALUES('TR-000010', 'Dentistry', 'Root Canal Procedure', 'PT-
000010', 'DR-000010', 'Sterilized Dental Instruments', 'RM-000010');
```

7. Diagnosis

```
INSERT INTO DIAGNOSIS VALUES('DX-000001', 'Hypertension', 'High blood pressure condition,
Emergency', 'PT-000001', 'DR-000001', TO_DATE ('11-01-2023', 'MM-DD-YYYY'));

INSERT INTO DIAGNOSIS VALUES('DX-000002', 'Type 2 Diabetes', 'Chronic condition affecting
blood sugar', 'PT-000002', 'DR-000002', TO_DATE ('11-02-2023', 'MM-DD-YYYY'));

INSERT INTO DIAGNOSIS VALUES('DX-000003', 'Asthma', 'Chronic respiratory condition,
Emergency', 'PT-000003', 'DR-000003', TO_DATE ('11-03-2023', 'MM-DD-YYYY'));

INSERT INTO DIAGNOSIS VALUES('DX-000004', 'Migraine', 'Recurring severe headaches', 'PT-
000004', 'DR-000004', TO_DATE ('11-04-2023', 'MM-DD-YYYY'));

INSERT INTO DIAGNOSIS VALUES('DX-000005', 'Osteoarthritis', 'Degenerative joint disease',
'PT-000005', 'DR-000005', TO_DATE ('11-05-2023', 'MM-DD-YYYY'));

INSERT INTO DIAGNOSIS VALUES('DX-000006', 'Anemia', 'Low hemoglobin levels', 'PT-000006',
'DR-000006', TO_DATE ('11-06-2023', 'MM-DD-YYYY'));

INSERT INTO DIAGNOSIS VALUES('DX-000007', 'Hyperthyroidism', 'Overactive thyroid gland',
'PT-000007', 'DR-000007', TO_DATE ('11-07-2023', 'MM-DD-YYYY'));

INSERT INTO DIAGNOSIS VALUES('DX-000008', 'Gastritis', 'Inflammation of the stomach
lining', 'PT-000008', 'DR-000008', TO_DATE ('11-08-2023', 'MM-DD-YYYY'));

INSERT INTO DIAGNOSIS VALUES('DX-000009', 'Chronic Sinusitis', 'Long-term inflammation of
sinuses', 'PT-000009', 'DR-000009', TO_DATE ('11-09-2023', 'MM-DD-YYYY'));

INSERT INTO DIAGNOSIS VALUES('DX-000010', 'Depression', 'Mental health condition causing
persistent sadness', 'PT-000010', 'DR-000010', TO_DATE ('11-10-2023', 'MM-DD-YYYY'));

INSERT INTO DIAGNOSIS VALUES('DX-000011', 'Hypertension', 'High blood pressure
condition', 'PT-000001', 'DR-000002', TO_DATE('11-01-2023', 'MM-DD-YYYY'));

INSERT INTO DIAGNOSIS VALUES('DX-000012', 'Type 2 Diabetes', 'Chronic condition
affecting blood sugar', 'PT-000002', 'DR-000003', TO_DATE('11-02-2023', 'MM-DD-
YYYY'));
```

8. Medical_Test

```
INSERT INTO MEDICAL_TEST VALUES('MT-000001', 'Blood Test', 'Normal', 'PT-000001', 'DR-
000001', TO_DATE ('11-10-2024', 'MM-DD-YYYY'), 'Fasting for 12 hours');

INSERT INTO MEDICAL_TEST VALUES('MT-000002', 'Lipid Profile ', 'High LDL levels', 'PT-
000002', 'DR-000002', TO_DATE ('11-11-2024', 'MM-DD-YYYY'), 'Avoid fatty foods before
test');

INSERT INTO MEDICAL_TEST VALUES('MT-000003', 'Chest X-Ray', 'Signs of mild infection',
'PT-000003', 'DR-000003', TO_DATE ('11-12-2024', 'MM-DD-YYYY'), 'Remove all metallic
items');

INSERT INTO MEDICAL_TEST VALUES('MT-000004', 'Urine Analysis', 'Presence of infection',
'PT-000004', 'DR-000004', TO_DATE ('11-13-2024', 'MM-DD-YYYY'), 'Midstream sample
required');

INSERT INTO MEDICAL_TEST VALUES('MT-000005', 'Thyroid Function Test', 'Elevated TSH
levels', 'PT-000005', 'DR-000005', TO_DATE ('11-14-2024', 'MM-DD-YYYY'), 'Take
medications as prescribed');

INSERT INTO MEDICAL_TEST VALUES('MT-000006', 'MRI Scan', 'Normal brain activity', 'PT-
000006', 'DR-000006', TO_DATE ('11-15-2024', 'MM-DD-YYYY'), 'No eating for 4 hours
prior');

INSERT INTO MEDICAL_TEST VALUES('MT-000007', 'Blood Glucose Test', 'Borderline diabetic
levels', 'PT-000007', 'DR-000007', TO_DATE ('11-16-2024', 'MM-DD-YYYY'), 'Fast for 8
hours');

INSERT INTO MEDICAL_TEST VALUES('MT-000008', 'Allergy Test', 'Positive for pollen
allergy', 'PT-000008', 'DR-000008', TO_DATE ('11-17-2024', 'MM-DD-YYYY'), 'Avoid
antihistamines for 3 days');

INSERT INTO MEDICAL_TEST VALUES('MT-000009', 'Electrocardiogram (ECG)', 'Irregular
heartbeat', 'PT-000009', 'DR-000009', TO_DATE ('11-18-2024', 'MM-DD-YYYY'), 'Relax before
test');

INSERT INTO MEDICAL_TEST VALUES('MT-000010', 'Bone Density Test', 'Osteopenia detected',
'PT-000010', 'DR-000010', TO_DATE ('11-19-2024', 'MM-DD-YYYY'), 'Calcium supplements to
be avoided');

INSERT INTO MEDICAL_TEST VALUES('MT-000011', 'Blood Test ', 'Normal', 'PT-000005', 'DR-
000011', TO_DATE ('11-10-2024', 'MM-DD-YYYY'), ' Fasting for 12 hours ');

INSERT INTO MEDICAL_TEST VALUES('MT-000012', 'Blood Test ', 'Anemia', 'PT-000007', 'DR-
000011', TO_DATE ('11-10-2024', 'MM-DD-YYYY'), ' Fasting for 12 hours ');
```

9. Pharmacy

```sql
INSERT INTO PHARMACY VALUES ('PH-001', 'Healthy Pharmacy', '123-456-7890', '123 Wellness
St.', '9 AM - 6 PM');

INSERT INTO PHARMACY VALUES ('PH-002', 'City Care Pharmacy', '234-567-8901', '456 City
Ave.', '8 AM - 8 PM');

INSERT INTO PHARMACY VALUES ('PH-003', 'CarePlus Pharmacy', '345-678-9012', '789 Care
Rd.', '10 AM - 5 PM');

INSERT INTO PHARMACY VALUES ('PH-004', 'Family Pharmacy', '456-789-0123', '101 Family
Blvd.', '9 AM - 7 PM');

INSERT INTO PHARMACY VALUES('PH-005', 'Greenleaf Pharmacy', '567-890-1234', '202
Greenleaf Ln.', '8 AM - 9 PM');

INSERT INTO PHARMACY VALUES ('PH-006', 'Sunshine Pharmacy', '678-901-2345', '303 Sunshine
St.', '9 AM - 6 PM');

INSERT INTO PHARMACY VALUES ('PH-007', 'CureWell Pharmacy', '789-012-3456', '404 Cure
Rd.', '8 AM - 7 PM');

INSERT INTO PHARMACY VALUES ('PH-008', 'PrimeCare Pharmacy', '890-123-4567', '505 Prime
Blvd.', '10 AM - 8 PM');

INSERT INTO PHARMACY VALUES('PH-009', 'QuickMed Pharmacy', '901-234-5678', '606 Quick
Lane', '7 AM - 9 PM');

INSERT INTO PHARMACY VALUES ('PH-010', 'MedExpress Pharmacy', '012-345-6789', '707 Med
Ave.', '9 AM - 5 PM');
```

10. Allergy

```
INSERT INTO ALLERGY VALUES ('AL-100001', 'Shellfish', 'Food allergy', 'Severe', 'Hives,
difficulty breathing', 'PT-000001' );

INSERT INTO ALLERGY VALUES ('AL-100002', 'Tree Nuts', 'Food allergy', 'Severe',
'Swelling, anaphylaxis', 'PT-000002');

INSERT INTO ALLERGY VALUES ('AL-100003', 'Gluten', 'Food allergy', 'Moderate', 'Abdominal
pain, bloating','PT-000003');

INSERT INTO ALLERGY VALUES ('AL-100004', 'Dairy', 'Food allergy', 'Moderate', 'Stomach
cramps, diarrhea', 'PT-000004');

INSERT INTO ALLERGY VALUES ('AL-100005', 'Aspirin', 'Drug allergy', 'Severe', 'Rash,
anaphylaxis', 'PT-000005');

INSERT INTO ALLERGY VALUES ('AL-100006', 'Amoxicillin', 'Drug allergy', 'High', 'Hives,
fever', 'PT-000006');

INSERT INTO ALLERGY VALUES ('AL-100007', 'Ibuprofen', 'Drug allergy', 'Moderate',
'Stomach upset, rash', 'PT-000007');

INSERT INTO ALLERGY VALUES ('AL-100008', 'Grass', 'Environmental allergy', 'Moderate',
'Sneezing, itchy eyes', 'PT-000008');

INSERT INTO ALLERGY VALUES ('AL-100009', 'Birch Pollen', 'Environmental allergy', 'High',
'Sneezing, runny nose', 'PT-000009');

INSERT INTO ALLERGY VALUES ('AL-100010', 'Bees', 'Environmental allergy', 'Severe',
'Anaphylaxis, swelling', 'PT-000010');

INSERT INTO ALLERGY VALUES ('AL-100011', 'Latex', 'Environmental allergy', 'Severe',
'Skin rash, difficulty breathing', 'PT-000011');
```

11. Medical_History

```
INSERT INTO MEDICAL_HISTORY VALUES ('MH-000001', 'PT-000001', TO_DATE ('01-05-2024, 'MM-
DD-YYYY'), 'Hypertension, Type 2 Diabetes', 'DR-000001', 70.5, 165.0, 'O+', '120/80',
5.6);
```

```
INSERT INTO MEDICAL_HISTORY VALUES ('MH-000002', 'PT-000002', TO_DATE ('02-02-2024, 'MM-
DD-YYYY'), ' Asthma, Allergic Rhinitis', 'DR-000002', 85.0, 175.0, 'A-', '130/85', 7.8);
```

```
INSERT INTO MEDICAL_HISTORY VALUES ('MH-000003', 'PT-000003', TO_DATE ('03-09-2024, 'MM-
DD-YYYY'), 'Obesity, Sleep Apnea' 'DR-000003', 130.0, 160.0, 'B+', '110/70', 4.8);
```

```
INSERT INTO MEDICAL_HISTORY VALUES ('MH-000004', 'PT-000004', TO_DATE ('04-05-2024, 'MM-
DD-YYYY'), 'Chronic Kidney Disease, Hypertension', 'DR-000004', 72.5, 168.0, 'AB-',
'115/75', 6.0);
```

```
INSERT INTO MEDICAL_HISTORY VALUES ('MH-000005', 'PT-000005', TO_DATE ('05-05-2024, 'MM-
DD-YYYY'), 'Coronary Artery Disease, Hyperlipidemia', 'DR-000001', 90.0, 180.0, 'O-',
'140/90', 7.2);
```

```
INSERT INTO MEDICAL_HISTORY VALUES ('MH-000006', 'PT-000006', TO_DATE ('06-02-2024, 'MM-
DD-YYYY'), 'Sickle Cell Anemia, Iron deficiency', 'DR-000006', 55.5, 150.0, 'A+',
'105/65', 5.1);
```

```
INSERT INTO MEDICAL_HISTORY VALUES ('MH-000007', 'PT-000007', TO_DATE ('07-08-2024, 'MM-
DD-YYYY'), 'Hyperthyroidism', 'DR-000007', 65.0, 170.0, 'B-', '125/80', 6.3);
```

```
INSERT INTO MEDICAL_HISTORY VALUES ('MH-000008', 'PT-000008', TO_DATE ('08-06-2024, 'MM-
DD-YYYY'), 'Throat Infection', 'DR-000003', 80.0, 176.0, 'AB+', '128/82', 8.0);
```

```
INSERT INTO MEDICAL_HISTORY VALUES ('MH-000009', 'PT-000009', TO_DATE ('09-02-2024, 'MM-
DD-YYYY'), 'Chronic Migraines', 'DR-000009', 68.0, 158.0, 'O+', '122/78', 6.8);
```

```
INSERT INTO MEDICAL_HISTORY VALUES ('MH-000010', 'PT-000010', TO_DATE ('10-05-2024, 'MM-
DD-YYYY'), 'Depression', 'DR-000010', 75.0, 172.0, 'A-', '118/76', 5.9);
```

12. Medication

```sql
INSERT INTO MEDICATION VALUES ('MED-00001', 'Aspirin', 'Pain reliever', 'Tablet', '500mg',
'PH-001');

INSERT INTO MEDICATION VALUES ('MED-00002', 'Amoxicillin', 'Antibiotic', 'Capsule',
'250mg', 'PH-002');

INSERT INTO MEDICATION VALUES ('MED-00003', 'Paracetamol', 'Pain reliever and fever
reducer', 'Tablet', '500mg', 'PH-003');

INSERT INTO MEDICATION VALUES ('MED-00004', 'Ibuprofen', 'Anti-inflammatory', 'Tablet',
'200mg', 'PH-004');

INSERT INTO MEDICATION VALUES ('MED-00005', 'Lisinopril', 'Blood pressure medication',
'Tablet', '10mg', 'PH-005');

INSERT INTO MEDICATION VALUES ('MED-00006', 'Simvastatin', 'Cholesterol-lowering
medication', 'Tablet', '20mg ', 'PH-006');

INSERT INTO MEDICATION VALUES ('MED-00007', 'Metformin', 'Diabetes medication', 'Tablet',
'500mg', 'PH-007');

INSERT INTO MEDICATION VALUES ('MED-00008', 'Cetirizine', 'Antihistamine', 'Tablet',
'10mg', 'PH-008');

INSERT INTO MEDICATION VALUES ('MED-00009', 'Albuterol', 'Asthma inhaler', 'Inhaler',
'90mcg', 'PH-009');

INSERT INTO MEDICATION VALUES ('MED-00010', 'Omeprazole', 'Acid reflux medication',
'Capsule', '20mg', 'PH-010');
```

13. Room

```
INSERT INTO ROOM VALUES ('RM-000001', 'Single', 1, 'JAN-00001', 'North Wing');

INSERT INTO ROOM VALUES ('RM-000002', 'Double', 2, 'JAN-00002', 'South Wing');

INSERT INTO ROOM VALUES ('RM-000003', 'Single', 1, 'JAN-00003', 'East Wing');

INSERT INTO ROOM VALUES ('RM-000004', 'VIP', 2, 'JAN-00004', 'West Wing');

INSERT INTO ROOM VALUES ('RM-000005', 'Triple', 3, 'JAN-00005', 'North Wing');

INSERT INTO ROOM VALUES ('RM-000006', 'Single', 1, 'JAN-00006', 'South Wing');

INSERT INTO ROOM VALUES ('RM-000007', 'Double', 2, 'JAN-00007', 'East Wing');

INSERT INTO ROOM VALUES ('RM-000008', 'Single', 1, 'JAN-00008', 'West Wing');

INSERT INTO ROOM VALUES ('RM-000009', 'Double', 2, 'JAN-00009', 'North Wing');

INSERT INTO ROOM VALUES ('RM-000010', 'Suite', 4, 'JAN-00010', 'Central Wing');
```

14. Medical_Equipment

```
INSERT INTO MEDICAL_EQUIPMENT VALUES ('EQ-000001', 'X-Ray Machine', 'Available', 'Cool,
Dry Place', 'MediEquip Ltd.’, 'RM-000001');

INSERT INTO MEDICAL_EQUIPMENT VALUES ('EQ-000002', 'MRI Scanner', 'Not Available',
'Magnetically Shielded', 'HealthTech Corp.', 'RM-000002');

INSERT INTO MEDICAL_EQUIPMENT VALUES ('EQ-000003', 'Ultrasound Machine', 'Available',
'Room Temperature', 'MediTech Supplies', 'RM-000003');

INSERT INTO MEDICAL_EQUIPMENT VALUES ('EQ-000004', 'Defibrillator', 'Available', 'Dry
Place', 'HeartSafe Inc.', 'RM-000004');

INSERT INTO MEDICAL_EQUIPMENT VALUES ('EQ-000005', 'CT Scanner', 'Not Available',
'Climate Controlled', 'ScanTech Ltd.', 'RM-000005');

INSERT INTO MEDICAL_EQUIPMENT VALUES ('EQ-000006', 'ECG Machine', 'Available', 'Cool, Dry
Place', 'CardioEquip Co.', 'RM-000006');

INSERT INTO MEDICAL_EQUIPMENT VALUES ('EQ-000007', 'Ventilator', 'Available', 'Clean
Area', 'LifeSupport Systems', 'RM-000007');

INSERT INTO MEDICAL_EQUIPMENT VALUES ('EQ-000008', 'Surgical Instruments', 'Available',
'Sterile Environment', 'SurgiTech', 'RM-000008');

INSERT INTO MEDICAL_EQUIPMENT VALUES ('EQ-000009', 'Infusion Pump', 'Not Available', 'Dry
Place', 'MediFlow Corp.', 'RM-000009');

INSERT INTO MEDICAL_EQUIPMENT VALUES ('EQ-000010', 'Patient Monitor', 'Available', 'Cool,
Dry Place', 'MonitorPlus', 'RM-000010');
```

15. Billing

```
INSERT INTO BILLING VALUES ('BILL-0001', TO_DATE('2024-10-01', 'YYYY-MM-DD'), 150.00,
'Credit Card', 'PT-000001', 'AP-000001');

INSERT INTO BILLING VALUES ('BILL-0002', TO_DATE('2024-10-02','YYYY-MM-DD'), 200.00,
'Cash', 'PT-000002', 'AP-000002');

INSERT INTO BILLING VALUES ('BILL-0003', TO_DATE ('2024-10-03','YYYY-MM-DD'), 120.00,
'Debit Card', 'PT-000003', 'AP-000003');

INSERT INTO BILLING VALUES ('BILL-0004', TO_DATE ('2024-10-04','YYYY-MM-DD'), 180.00,
'Credit Card', 'PT-000004', 'AP-000004');

INSERT INTO BILLING VALUES ('BILL-0005', TO_DATE ('2024-10-05','YYYY-MM-DD'), 250.00,
'Cash', 'PT-000005', 'AP-000005');

INSERT INTO BILLING VALUES ('BILL-0006', TO_DATE ('2024-10-06','YYYY-MM-DD'), 220.00,
'Debit Card', 'PT-000006', 'AP-000006');

INSERT INTO BILLING VALUES ('BILL-0007', TO_DATE ('2024-10-07','YYYY-MM-DD'), 140.00,
'Credit Card', 'PT-000007', 'AP-000007');

INSERT INTO BILLING VALUES ('BILL-0008', TO_DATE ('2024-10-08','YYYY-MM-DD'), 210.00,
'Cash', 'PT-000008', 'AP-000008');

INSERT INTO BILLING VALUES ('BILL-0009', TO_DATE ('2024-10-09','YYYY-MM-DD'), 130.00,
'Debit Card', 'PT-000009', 'AP-000009');

INSERT INTO BILLING VALUES ('BILL-0010', TO_DATE ('2024-10-10','YYYY-MM-DD'), 160.00,
'Credit Card', 'PT-000010', 'AP-000010');

INSERT INTO BILLING VALUES ('BILL-0011', TO_DATE('2024-04-05', 'YYYY-MM-DD'), 200.00,
'Credit Card', 'PT-000001', 'AP-000011');
```

16. Prescription

```
INSERT INTO PRESCRIPTION VALUES('P12345', 'Loratadine for allergy relief', 'PT-000001',
'DR-000001', TO_DATE('2024-11-15', 'YYYY-MM-DD'));

INSERT INTO PRESCRIPTION VALUES('P12346', 'Ibuprofen for pain relief', 'PT-000002', 'DR-
000002', TO_DATE('2024-11-16', 'YYYY-MM-DD'));

INSERT INTO PRESCRIPTION VALUES('P12347', 'Prednisone for inflammation', 'PT-000003',
'DR-000003', TO_DATE('2024-11-17', 'YYYY-MM-DD'));

INSERT INTO PRESCRIPTION VALUES('P12348', 'Aspirin for headache', 'PT-000004', 'DR-
000004', TO_DATE('2024-11-18', 'YYYY-MM-DD'));

INSERT INTO PRESCRIPTION VALUES('P12349', 'Cetirizine for allergy symptoms', 'PT-000005',
'DR-000005', TO_DATE('2024-11-19', 'YYYY-MM-DD'));

INSERT INTO PRESCRIPTION VALUES('P12350', 'Amoxicillin for throat infection', 'PT-
000006', 'DR-000003', TO_DATE('2024-11-20', 'YYYY-MM-DD'));

INSERT INTO PRESCRIPTION VALUES('P12351', 'Omeprazole for acid reflux', 'PT-000007', 'DR-
000007', TO_DATE('2024-11-21', 'YYYY-MM-DD'));

INSERT INTO PRESCRIPTION VALUES('P12352', 'Metformin for diabetes', 'PT-000008', 'DR-
000008', TO_DATE('2024-11-22', 'YYYY-MM-DD'));

INSERT INTO PRESCRIPTION VALUES('P12353', 'Hydrochlorothiazide for high blood pressure',
'PT-000009', 'DR-000009', TO_DATE('2024-11-23', 'YYYY-MM-DD'));

INSERT INTO PRESCRIPTION VALUES('P12354', 'Simvastatin for cholesterol', 'PT-000010',
'DR-000010', TO_DATE('2024-11-24', 'YYYY-MM-DD'));
```

17. Belongs_To2

```
INSERT INTO BELONGS_TO2 VALUES('NS-000001', 'DP-000001', TO_DATE ('12-05-2021', 'MM-DD-
YYYY'), NULL);

INSERT INTO BELONGS_TO2 VALUES('NS-000002', 'DP-000002', TO_DATE ('10-10-2022', 'MM-DD-
YYYY'), NULL);

INSERT INTO BELONGS_TO2 VALUES('NS-000003', 'DP-000003', TO_DATE ('02-28-2019', 'MM-DD-
YYYY'), NULL);

INSERT INTO BELONGS_TO2 VALUES('NS-000004', 'DP-000004', TO_DATE ('12-30-2023', 'MM-DD-
YYYY'), NULL);

INSERT INTO BELONGS_TO2 VALUES('NS-000005', 'DP-000008', TO_DATE ('03-06-2021', 'MM-DD-
YYYY'), NULL);

INSERT INTO BELONGS_TO2 VALUES('NS-000006', 'DP-000007', TO_DATE ('08-11-2022', 'MM-DD-
YYYY'), NULL);

INSERT INTO BELONGS_TO2 VALUES('NS-000007', 'DP-000009', TO_DATE ('11-22-2023', 'MM-DD-
YYYY'), NULL);

INSERT INTO BELONGS_TO2 VALUES('NS-000008', 'DP-000010', TO_DATE ('05-08-2023', 'MM-DD-
YYYY'), NULL);

INSERT INTO BELONGS_TO2 VALUES('NS-000009', 'DP-000007', TO_DATE ('08-26-2023', 'MM-DD-
YYYY'), NULL);

INSERT INTO BELONGS_TO2 VALUES('NS-000010', 'DP-000005', TO_DATE ('09-15-2024', 'MM-DD-
YYYY'), NULL);
```

18. Services

```sql
INSERT INTO SERVICES VALUES('NS-000001', 'PT-000005');

INSERT INTO SERVICES VALUES('NS-000002', 'PT-000003');

INSERT INTO SERVICES VALUES('NS-000003', 'PT-000002');

INSERT INTO SERVICES VALUES('NS-000004', 'PT-000006');

INSERT INTO SERVICES VALUES('NS-000005', 'DP-000008');

INSERT INTO SERVICES VALUES('NS-000006', 'PT-000004');

INSERT INTO SERVICES VALUES('NS-000007', 'PT-000009');

INSERT INTO SERVICES VALUES('NS-000008', 'PT-000010');

INSERT INTO SERVICES VALUES('NS-000009', 'PT-000008');

INSERT INTO SERVICES VALUES('NS-000010', NULL);
```

19. Visits

```
INSERT INTO VISITS VALUES('PT-000001', 'DR-000001', TO_DATE('10-01-2024', 'MM-DD-YYYY') ,
'Regular Checkup');

INSERT INTO VISITS VALUES('PT-000002', 'DR-000002, TO_DATE('01-03-2024', 'MM-DD-YYYY') ,
'Recurring Asthma Attacks');

INSERT INTO VISITS VALUES('PT-000003', 'DR-000003', TO_DATE('02-22-2024', 'MM-DD-YYYY') ,
'Flu virus');

INSERT INTO VISITS VALUES('PT-000004', 'DR-000004', TO_DATE('11-11-2024', 'MM-DD-YYYY') ,
'Kidney Transplant Pre-Surgery Evaluations');

INSERT INTO VISITS VALUES('PT-000005', 'DR-000005', TO_DATE('11-03-2024', 'MM-DD-YYYY') ,
'Regular Checkup');

INSERT INTO VISITS VALUES('PT-000006', 'DR-000003', TO_DATE('11-20-2024', 'MM-DD-YYYY'),
'Throat Infection');

INSERT INTO VISITS VALUES('PT-000007', 'DR-000007', TO_DATE('05-05-2024', 'MM-DD-YYYY')
'Regular Checkup');

INSERT INTO VISITS VALUES('PT-000008', 'DR-000008' TO_DATE('02-09-2024', 'MM-DD-YYYY')
'CT Scan After Car Accident');

INSERT INTO VISITS VALUES('PT-000009', 'DR-000009', TO_DATE('10-29-2024', 'MM-DD-YYYY')
'Pregnancy Checkup');

INSERT INTO VISITS VALUES('PT-000010', 'DR-000010', TO_DATE('09-06-2024', 'MM-DD-YYYY')
'Suicide Attempt');
```

20. Requires

```
INSERT INTO REQUIRES VALUES('MT-000001', 'DX-000001');

INSERT INTO REQUIRES VALUES('MT-000002', 'DX-000002');

INSERT INTO REQUIRES VALUES('MT-000003', 'DX-000003');

INSERT INTO REQUIRES VALUES('MT-000004', 'DX-000004');

INSERT INTO REQUIRES VALUES('MT-000005', 'DX-000005');

INSERT INTO REQUIRES VALUES('MT-000006', 'DX-000006');

INSERT INTO REQUIRES VALUES('MT-000007', 'DX-000007');

INSERT INTO REQUIRES VALUES('MT-000008', 'DX-000008');

INSERT INTO REQUIRES VALUES('MT-000009', 'DX-000009');

INSERT INTO REQUIRES VALUES('MT-000010', 'DX-000010');
```

21. Contains1

```
INSERT INTO CONTAINS1 VALUES ('P12345', 'MED-00001');

INSERT INTO CONTAINS1 VALUES ('P12346', 'MED-00006');

INSERT INTO CONTAINS1 VALUES ('P12347', 'MED-00003');

INSERT INTO CONTAINS1 VALUES ('P12348', 'MED-00004');

INSERT INTO CONTAINS1 VALUES ('P12349', 'MED-00005');

INSERT INTO CONTAINS1 VALUES ('P12350', 'MED-00002');

INSERT INTO CONTAINS1 VALUES ('P12351', 'MED-00007');

INSERT INTO CONTAINS1 VALUES ('P12352', 'MED-00008');

INSERT INTO CONTAINS1 VALUES ('P12353', 'MED-00009');

INSERT INTO CONTAINS1 VALUES ('P12354', 'MED-00010');
```

22. Treats

```
INSERT INTO TREATS VALUES ('DR-000001', 'PT-000001');
INSERT INTO TREATS VALUES ('DR-000002', 'PT-000002');
INSERT INTO TREATS VALUES ('DR-000003', 'PT-000003’);
INSERT INTO TREATS VALUES ('DR-000004', 'PT-000004');
INSERT INTO TREATS VALUES ('DR-000005', 'PT-000005');
INSERT INTO TREATS VALUES ('DR-000006', 'PT-000006');
INSERT INTO TREATS VALUES ('DR-000007', 'PT-000007');
INSERT INTO TREATS VALUES ('DR-000008', 'PT-000008');
INSERT INTO TREATS VALUES ('DR-000009', 'PT-000009');
INSERT INTO TREATS VALUES ('DR-000010', 'PT-000010');
```

23. Undergoes

```
INSERT INTO UNDERGOES VALUES('DR-000001', 'PT-000001', TO_DATE('10-05-2024', 'MM-DD-
YYYY'));

INSERT INTO UNDERGOES VALUES('DR-000002', 'PT-000002', TO_DATE('11-02-2024', 'MM-DD-
YYYY'));

INSERT INTO UNDERGOES VALUES('DR-000003', 'PT-000003', TO_DATE('09-01-2024', 'MM-DD-
YYYY'));

INSERT INTO UNDERGOES VALUES('DR-000004', 'PT-000004', TO_DATE('04-26-2024', 'MM-DD-
YYYY'));

INSERT INTO UNDERGOES VALUES('DR-000005', 'PT-000005', TO_DATE('07-05-2024', 'MM-DD-
YYYY'));

INSERT INTO UNDERGOES VALUES('DR-000006', 'PT-000006', TO_DATE('03-09-2024', 'MM-DD-
YYYY'));

INSERT INTO UNDERGOES VALUES('DR-000007', 'PT-000007', TO_DATE('08-30-2024', 'MM-DD-
YYYY'));

INSERT INTO UNDERGOES VALUES('DR-000008', 'PT-000008', TO_DATE('02-01-2024', 'MM-DD-
YYYY'));

INSERT INTO UNDERGOES VALUES('DR-000009', 'PT-000009', TO_DATE('01-27-2024', 'MM-DD-
YYYY'));

INSERT INTO UNDERGOES VALUES('DR-000010', 'PT-000010', TO_DATE('11-23-2024', 'MM-DD-
YYYY'));
```

24. Doctor_Phone_Number

```
INSERT INTO DOCTOR_PHONE_NUMBER VALUES ('DR-000001', '03-973572');
INSERT INTO DOCTOR_PHONE_NUMBER VALUES ('DR-000002', '71-926253');
INSERT INTO DOCTOR_PHONE_NUMBER VALUES ('DR-000003', '03-875345′);
INSERT INTO DOCTOR_PHONE_NUMBER VALUES ('DR-000004', '76-624961');
INSERT INTO DOCTOR_PHONE_NUMBER VALUES ('DR-000005', '71-243342');
INSERT INTO DOCTOR_PHONE_NUMBER VALUES ('DR-000006', '71-986427');
INSERT INTO DOCTOR_PHONE_NUMBER VALUES ('DR-000007', '71-547643');
INSERT INTO DOCTOR_PHONE_NUMBER VALUES ('DR-000008', '03-215468');
INSERT INTO DOCTOR_PHONE_NUMBER VALUES ('DR-000009', '03-886995');
INSERT INTO DOCTOR_PHONE_NUMBER VALUES ('DR-000010', '71-990543');
INSERT INTO DOCTOR_PHONE_NUMBER VALUES ('DR-000011', '76-994631');
```

25. Nurse_Phone_Number

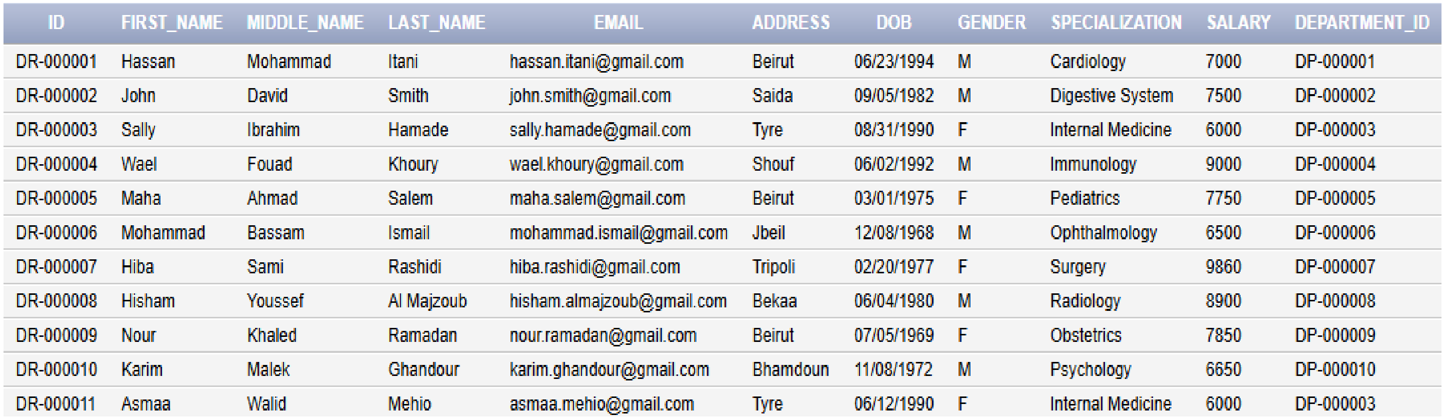
```
INSERT INTO NURSE_PHONE_NUMBER VALUES ('NS-000001', '03-765452');
INSERT INTO NURSE_PHONE_NUMBER VALUES ('NS-000002', '71-054346');
INSERT INTO NURSE_PHONE_NUMBER VALUES ('NS-000003', '03-123123');
INSERT INTO NURSE_PHONE_NUMBER VALUES ('NS-000004', '76-178965');
INSERT INTO NURSE_PHONE_NUMBER VALUES ('NS-000005', '71-232944');
INSERT INTO NURSE_PHONE_NUMBER VALUES ('NS-000006', '71-880887');
INSERT INTO NURSE_PHONE_NUMBER VALUES ('NS-000007', '71-504672');
INSERT INTO NURSE_PHONE_NUMBER VALUES ('NS-000008', '03-987789');
INSERT INTO NURSE_PHONE_NUMBER VALUES ('NS-000009', '03-868959');
INSERT INTO NURSE_PHONE_NUMBER VALUES ('NS-000010', '71-345990');
```

26. Patient_Phone_Number

```
INSERT INTO PATIENT_PHONE_NUMBER VALUES('PT-000001', '03-254678');
INSERT INTO PATIENT_PHONE_NUMBER VALUES('PT-000002', '71-121167');
INSERT INTO PATIENT_PHONE_NUMBER VALUES('PT-000003', '03-321654');
INSERT INTO PATIENT_PHONE_NUMBER VALUES('PT-000004', '76-965345');
INSERT INTO PATIENT_PHONE_NUMBER VALUES('PT-000005', '71-442233');
INSERT INTO PATIENT_PHONE_NUMBER VALUES('PT-000006', '71-870230');
INSERT INTO PATIENT_PHONE_NUMBER VALUES('PT-000007', '71-405105');
INSERT INTO PATIENT_PHONE_NUMBER VALUES('PT-000008', '03-878989');
INSERT INTO PATIENT_PHONE_NUMBER VALUES('PT-000009', '03-303456');
INSERT INTO PATIENT_PHONE_NUMBER VALUES('PT-000010', '71-399653');
```

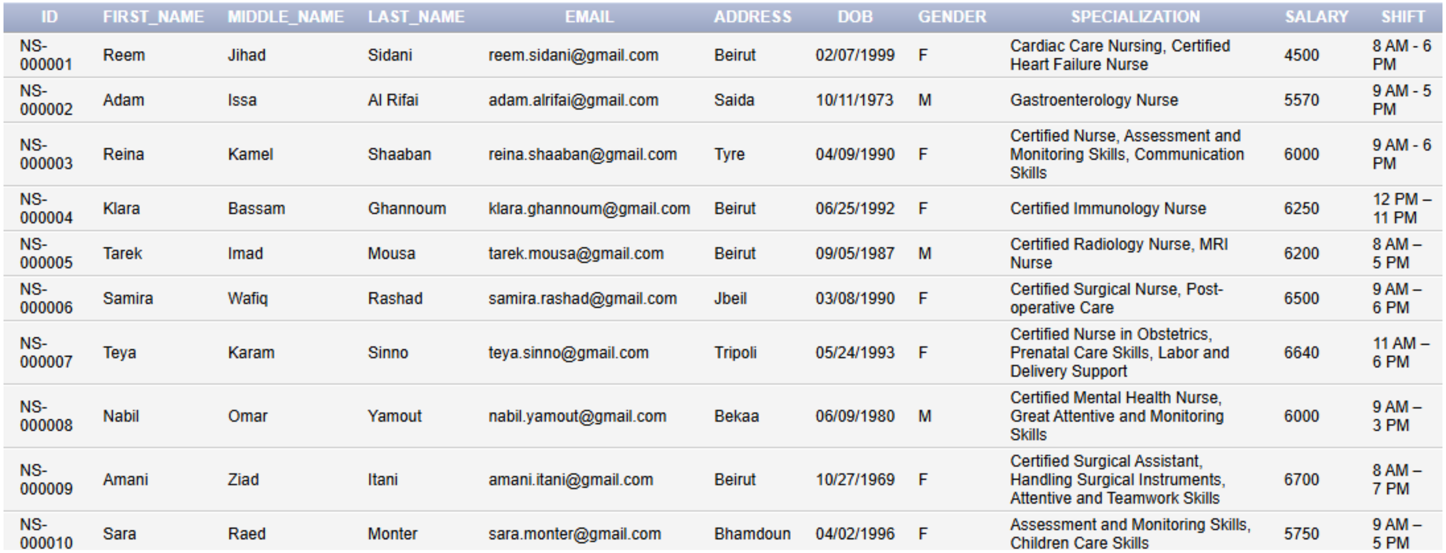
XIII. Final Tables State:

1. Doctor



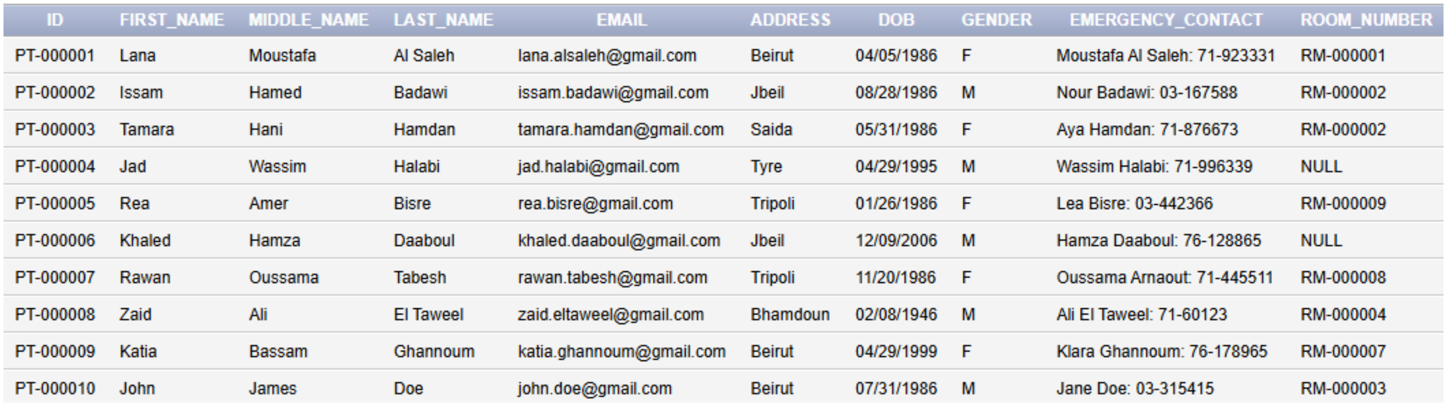
ID	FIRST_NAME	MIDDLE_NAME	LAST_NAME	EMAIL	ADDRESS	DOB	GENDER	SPECIALIZATION	SALARY	DEPARTMENT_ID
DR-000001	Hassan	Mohammad	Itani	hassan.itani@gmail.com	Beirut	06/23/1994	M	Cardiology	7000	DP-000001
DR-000002	John	David	Smith	john.smith@gmail.com	Saida	09/05/1982	M	Digestive System	7500	DP-000002
DR-000003	Sally	Ibrahim	Hamade	sally.hamade@gmail.com	Tyre	08/31/1990	F	Internal Medicine	6000	DP-000003
DR-000004	Wael	Fouad	Khoury	wael.khoury@gmail.com	Shouf	06/02/1992	M	Immunology	9000	DP-000004
DR-000005	Maha	Ahmad	Salem	maha.salem@gmail.com	Beirut	03/01/1975	F	Pediatrics	7750	DP-000005
DR-000006	Mohammad	Bassam	Ismail	mohammad.ismail@gmail.com	Jbeil	12/08/1968	M	Ophthalmology	6500	DP-000006
DR-000007	Hiba	Sami	Rashidi	hiba.rashidi@gmail.com	Tripoli	02/20/1977	F	Surgery	9860	DP-000007
DR-000008	Hisham	Youssef	Al Majzoub	hisham.almajzoub@gmail.com	Bekaa	06/04/1980	M	Radiology	8900	DP-000008
DR-000009	Nour	Khaled	Ramadan	nour.ramadan@gmail.com	Beirut	07/05/1969	F	Obstetrics	7850	DP-000009
DR-000010	Karim	Malek	Ghandour	karim.ghandour@gmail.com	Bhamdoun	11/08/1972	M	Psychology	6650	DP-000010
DR-000011	Asmaa	Walid	Mehio	asmaa.mehio@gmail.com	Tyre	06/12/1990	F	Internal Medicine	6000	DP-000003

2. Nurse



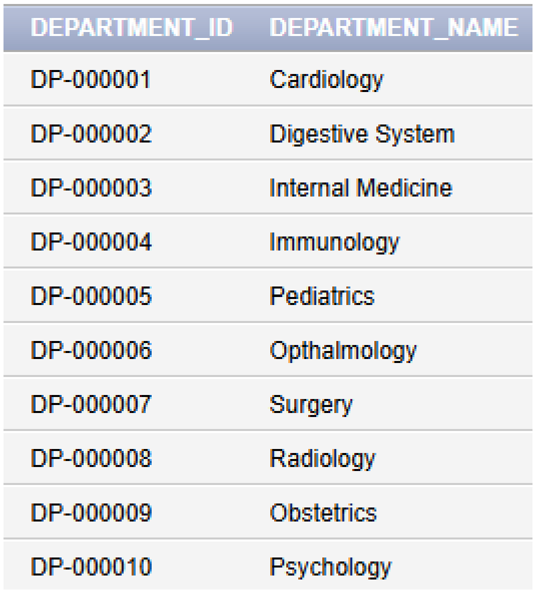
ID	FIRST_NAME	MIDDLE_NAME	LAST_NAME	EMAIL	ADDRESS	DOB	GENDER	SPECIALIZATION	SALARY	SHIFT
NS-000001	Reem	Jihad	Sidani	reem.sidani@gmail.com	Beirut	02/07/1999	F	Cardiac Care Nursing, Certified Heart Failure Nurse	4500	8 AM - 6 PM
NS-000002	Adam	Issa	Al Rifai	adam.alrifai@gmail.com	Saida	10/11/1973	M	Gastroenterology Nurse	5570	9 AM - 5 PM
NS-000003	Reina	Kamel	Shaaban	reina.shaaban@gmail.com	Tyre	04/09/1990	F	Certified Nurse, Assessment and Monitoring Skills, Communication Skills	6000	9 AM - 6 PM
NS-000004	Klara	Bassam	Ghannoum	klara.ghannoum@gmail.com	Beirut	06/25/1992	F	Certified Immunology Nurse	6250	12 PM – 11 PM
NS-000005	Tarek	Imad	Mousa	tarek.mousa@gmail.com	Beirut	09/05/1987	M	Certified Radiology Nurse, MRI Nurse	6200	8 AM – 5 PM
NS-000006	Samira	Wafiq	Rashad	samira.rashad@gmail.com	Jbeil	03/08/1990	F	Certified Surgical Nurse, Post-operative Care	6500	9 AM – 6 PM
NS-000007	Teya	Karam	Sinno	teya.sinno@gmail.com	Tripoli	05/24/1993	F	Certified Nurse in Obstetrics, Prenatal Care Skills, Labor and Delivery Support	6640	11 AM – 6 PM
NS-000008	Nabil	Omar	Yamout	nabil.yamout@gmail.com	Bekaa	06/09/1980	M	Certified Mental Health Nurse, Great Attentive and Monitoring Skills	6000	9 AM – 3 PM
NS-000009	Amani	Ziad	Itani	amani.itani@gmail.com	Beirut	10/27/1969	F	Certified Surgical Assistant, Handling Surgical Instruments, Attentive and Teamwork Skills	6700	8 AM – 7 PM
NS-000010	Sara	Raed	Monter	sara.monter@gmail.com	Bhamdoun	04/02/1996	F	Assessment and Monitoring Skills, Children Care Skills	5750	9 AM – 5 PM

3. Patient



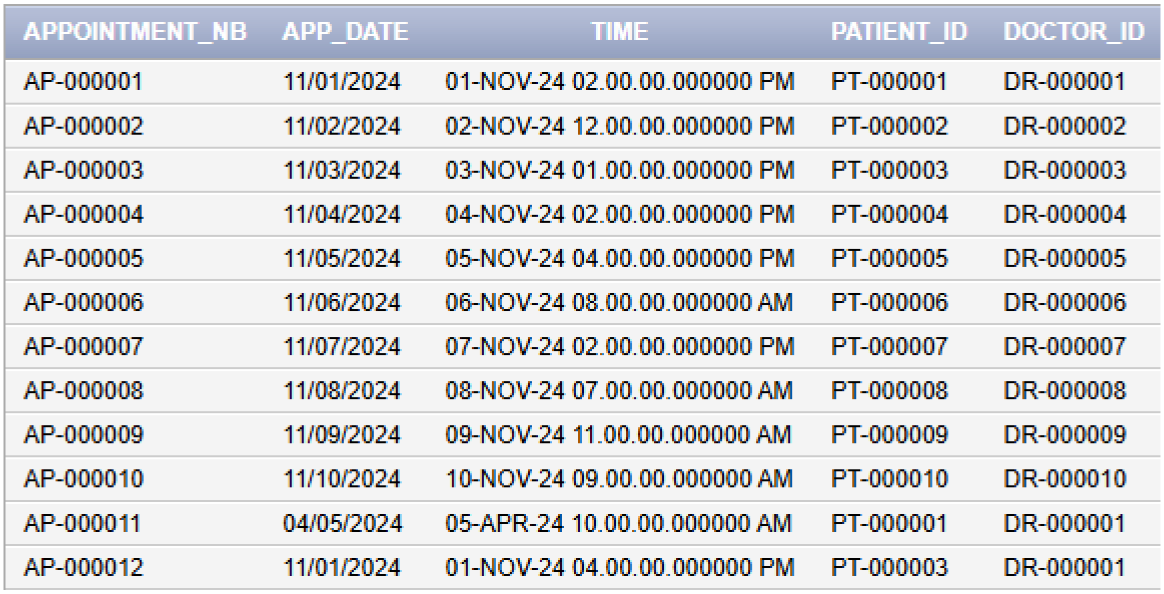
ID	FIRST_NAME	MIDDLE_NAME	LAST_NAME	EMAIL	ADDRESS	DOB	GENDER	EMERGENCY_CONTACT	ROOM_NUMBER
PT-000001	Lana	Moustafa	Al Saleh	lana.alsaleh@gmail.com	Beirut	04/05/1986	F	Moustafa Al Saleh: 71-923331	RM-000001
PT-000002	Issam	Hamed	Badawi	issam.badawi@gmail.com	Jbeil	08/28/1986	M	Nour Badawi: 03-167588	RM-000002
PT-000003	Tamara	Hani	Hamdan	tamara.hamdan@gmail.com	Saida	05/31/1986	F	Aya Hamdan: 71-876673	RM-000002
PT-000004	Jad	Wassim	Halabi	jad.halabi@gmail.com	Tyre	04/29/1995	M	Wassim Halabi: 71-996339	NULL
PT-000005	Rea	Amer	Bisre	rea.bisre@gmail.com	Tripoli	01/26/1986	F	Lea Bisre: 03-442366	RM-000009
PT-000006	Khaled	Hamza	Daaboul	khaled.daaboul@gmail.com	Jbeil	12/09/2006	M	Hamza Daaboul: 76-128865	NULL
PT-000007	Rawan	Oussama	Tabesh	rawan.tabesh@gmail.com	Tripoli	11/20/1986	F	Oussama Arnaout: 71-445511	RM-000008
PT-000008	Zaid	Ali	El Taweel	zaid.eltaweel@gmail.com	Bhamdoun	02/08/1946	M	Ali El Taweel: 71-60123	RM-000004
PT-000009	Katia	Bassam	Ghannoum	katia.ghannoum@gmail.com	Beirut	04/29/1999	F	Klara Ghannoum: 76-178965	RM-000007
PT-000010	John	James	Doe	john.doe@gmail.com	Beirut	07/31/1986	M	Jane Doe: 03-315415	RM-000003

4. Department



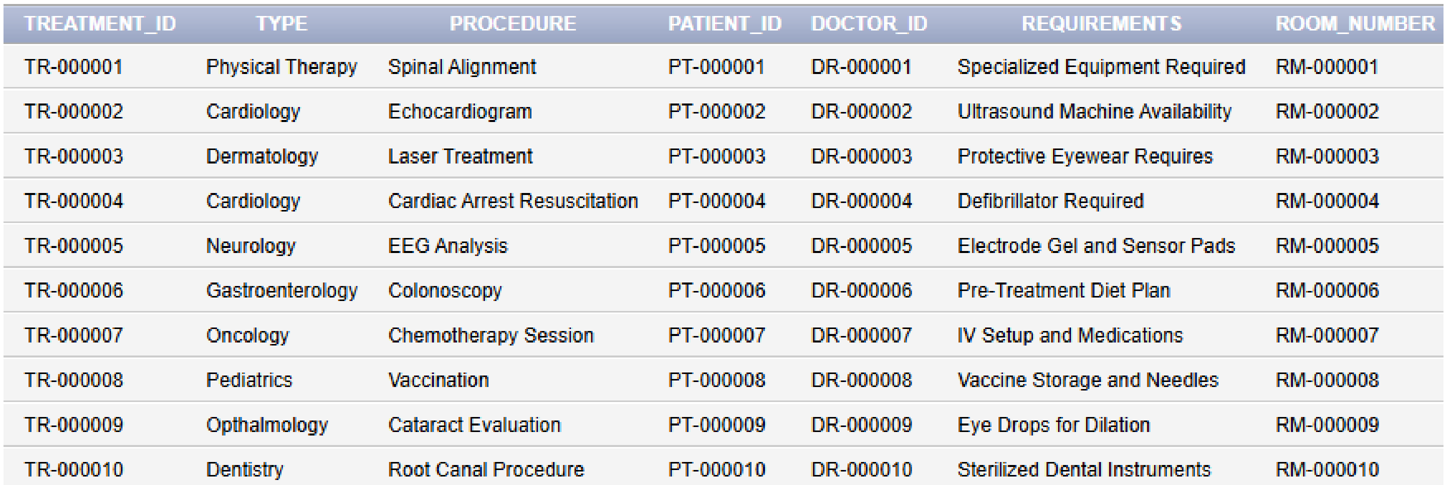
DEPARTMENT_ID	DEPARTMENT_NAME
DP-000001	Cardiology
DP-000002	Digestive System
DP-000003	Internal Medicine
DP-000004	Immunology
DP-000005	Pediatrics
DP-000006	Opthalmology
DP-000007	Surgery
DP-000008	Radiology
DP-000009	Obstetrics
DP-000010	Psychology

5. Appointment



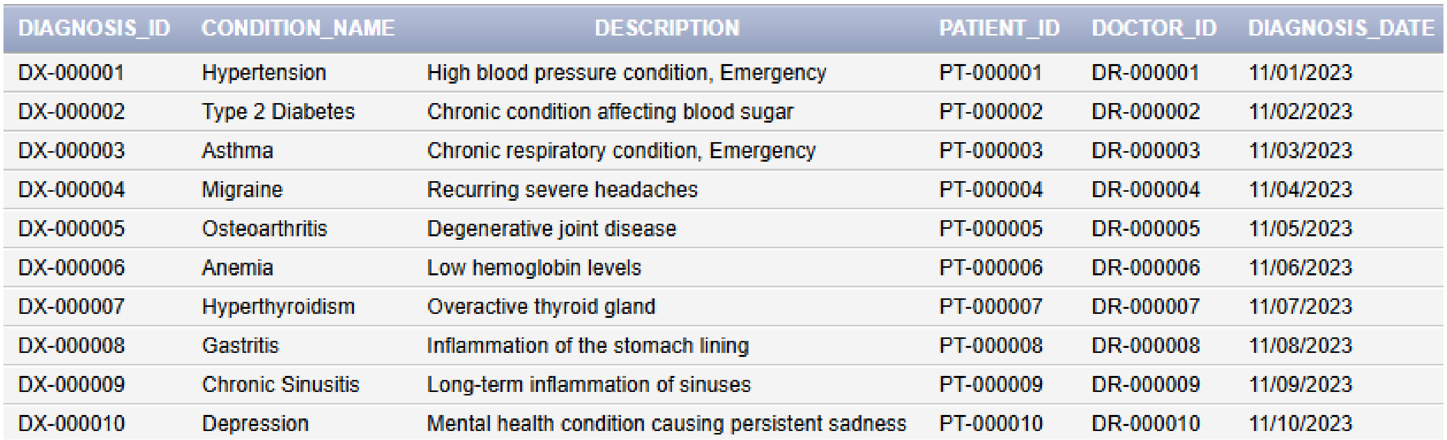
APPOINTMENT_NB	APP_DATE	TIME	PATIENT_ID	DOCTOR_ID
AP-000001	11/01/2024	01-NOV-24 02.00.00.000000 PM	PT-000001	DR-000001
AP-000002	11/02/2024	02-NOV-24 12.00.00.000000 PM	PT-000002	DR-000002
AP-000003	11/03/2024	03-NOV-24 01.00.00.000000 PM	PT-000003	DR-000003
AP-000004	11/04/2024	04-NOV-24 02.00.00.000000 PM	PT-000004	DR-000004
AP-000005	11/05/2024	05-NOV-24 04.00.00.000000 PM	PT-000005	DR-000005
AP-000006	11/06/2024	06-NOV-24 08.00.00.000000 AM	PT-000006	DR-000006
AP-000007	11/07/2024	07-NOV-24 02.00.00.000000 PM	PT-000007	DR-000007
AP-000008	11/08/2024	08-NOV-24 07.00.00.000000 AM	PT-000008	DR-000008
AP-000009	11/09/2024	09-NOV-24 11.00.00.000000 AM	PT-000009	DR-000009
AP-000010	11/10/2024	10-NOV-24 09.00.00.000000 AM	PT-000010	DR-000010
AP-000011	04/05/2024	05-APR-24 10.00.00.000000 AM	PT-000001	DR-000001
AP-000012	11/01/2024	01-NOV-24 04.00.00.000000 PM	PT-000003	DR-000001

6. Treatment



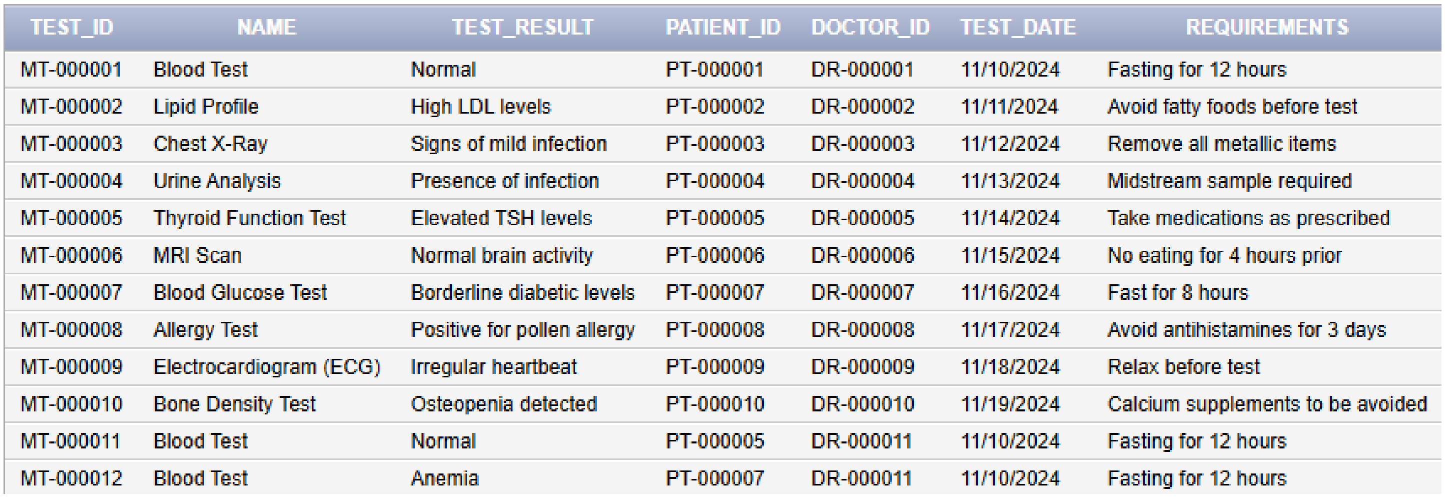
TREATMENT_ID	TYPE	PROCEDURE	PATIENT_ID	DOCTOR_ID	REQUIREMENTS	ROOM_NUMBER
TR-000001	Physical Therapy	Spinal Alignment	PT-000001	DR-000001	Specialized Equipment Required	RM-000001
TR-000002	Cardiology	Echocardiogram	PT-000002	DR-000002	Ultrasound Machine Availability	RM-000002
TR-000003	Dermatology	Laser Treatment	PT-000003	DR-000003	Protective Eyewear Requires	RM-000003
TR-000004	Cardiology	Cardiac Arrest Resuscitation	PT-000004	DR-000004	Defibrillator Required	RM-000004
TR-000005	Neurology	EEG Analysis	PT-000005	DR-000005	Electrode Gel and Sensor Pads	RM-000005
TR-000006	Gastroenterology	Colonoscopy	PT-000006	DR-000006	Pre-Treatment Diet Plan	RM-000006
TR-000007	Oncology	Chemotherapy Session	PT-000007	DR-000007	IV Setup and Medications	RM-000007
TR-000008	Pediatrics	Vaccination	PT-000008	DR-000008	Vaccine Storage and Needles	RM-000008
TR-000009	Opthalmology	Cataract Evaluation	PT-000009	DR-000009	Eye Drops for Dilation	RM-000009
TR-000010	Dentistry	Root Canal Procedure	PT-000010	DR-000010	Sterilized Dental Instruments	RM-000010

7. Diagnosis



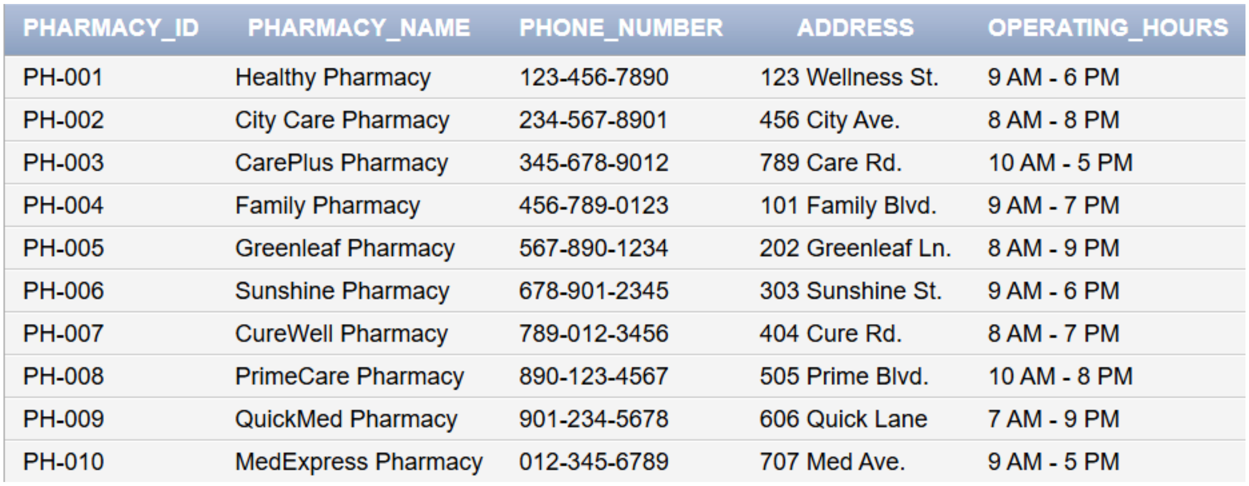
DIAGNOSIS_ID	CONDITION_NAME	DESCRIPTION	PATIENT_ID	DOCTOR_ID	DIAGNOSIS_DATE
DX-000001	Hypertension	High blood pressure condition, Emergency	PT-000001	DR-000001	11/01/2023
DX-000002	Type 2 Diabetes	Chronic condition affecting blood sugar	PT-000002	DR-000002	11/02/2023
DX-000003	Asthma	Chronic respiratory condition, Emergency	PT-000003	DR-000003	11/03/2023
DX-000004	Migraine	Recurring severe headaches	PT-000004	DR-000004	11/04/2023
DX-000005	Osteoarthritis	Degenerative joint disease	PT-000005	DR-000005	11/05/2023
DX-000006	Anemia	Low hemoglobin levels	PT-000006	DR-000006	11/06/2023
DX-000007	Hyperthyroidism	Overactive thyroid gland	PT-000007	DR-000007	11/07/2023
DX-000008	Gastritis	Inflammation of the stomach lining	PT-000008	DR-000008	11/08/2023
DX-000009	Chronic Sinusitis	Long-term inflammation of sinuses	PT-000009	DR-000009	11/09/2023
DX-000010	Depression	Mental health condition causing persistent sadness	PT-000010	DR-000010	11/10/2023

8. Medical_Test



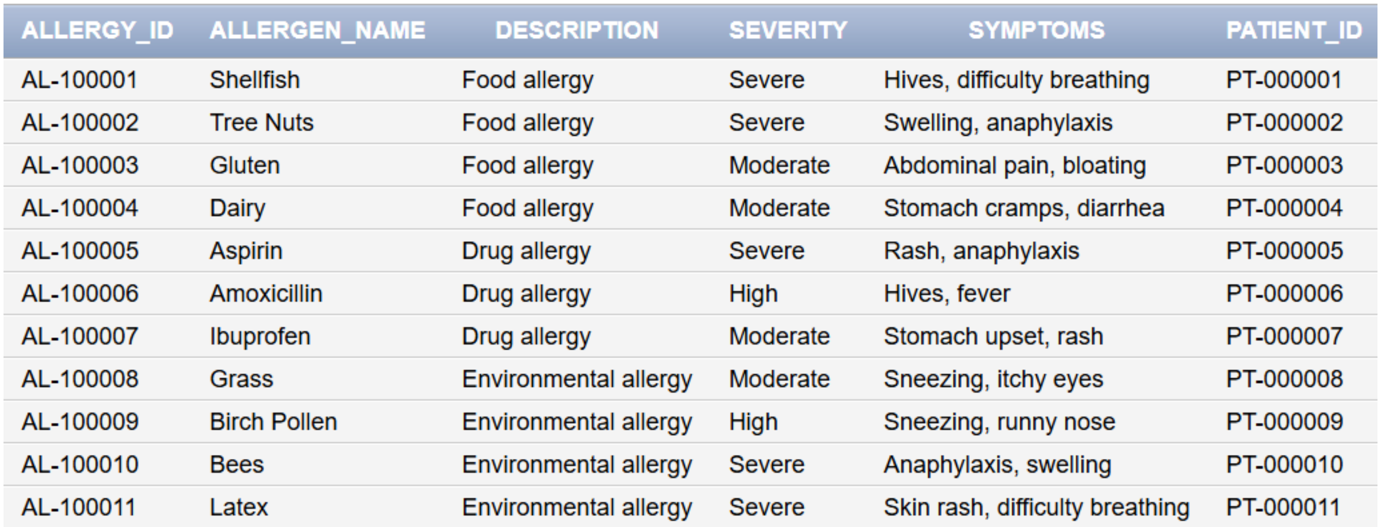
TEST_ID	NAME	TEST_RESULT	PATIENT_ID	DOCTOR_ID	TEST_DATE	REQUIREMENTS
MT-000001	Blood Test	Normal	PT-000001	DR-000001	11/10/2024	Fasting for 12 hours
MT-000002	Lipid Profile	High LDL levels	PT-000002	DR-000002	11/11/2024	Avoid fatty foods before test
MT-000003	Chest X-Ray	Signs of mild infection	PT-000003	DR-000003	11/12/2024	Remove all metallic items
MT-000004	Urine Analysis	Presence of infection	PT-000004	DR-000004	11/13/2024	Midstream sample required
MT-000005	Thyroid Function Test	Elevated TSH levels	PT-000005	DR-000005	11/14/2024	Take medications as prescribed
MT-000006	MRI Scan	Normal brain activity	PT-000006	DR-000006	11/15/2024	No eating for 4 hours prior
MT-000007	Blood Glucose Test	Borderline diabetic levels	PT-000007	DR-000007	11/16/2024	Fast for 8 hours
MT-000008	Allergy Test	Positive for pollen allergy	PT-000008	DR-000008	11/17/2024	Avoid antihistamines for 3 days
MT-000009	Electrocardiogram (ECG)	Irregular heartbeat	PT-000009	DR-000009	11/18/2024	Relax before test
MT-000010	Bone Density Test	Osteopenia detected	PT-000010	DR-000010	11/19/2024	Calcium supplements to be avoided
MT-000011	Blood Test	Normal	PT-000005	DR-000011	11/10/2024	Fasting for 12 hours
MT-000012	Blood Test	Anemia	PT-000007	DR-000011	11/10/2024	Fasting for 12 hours

9. Pharmacy



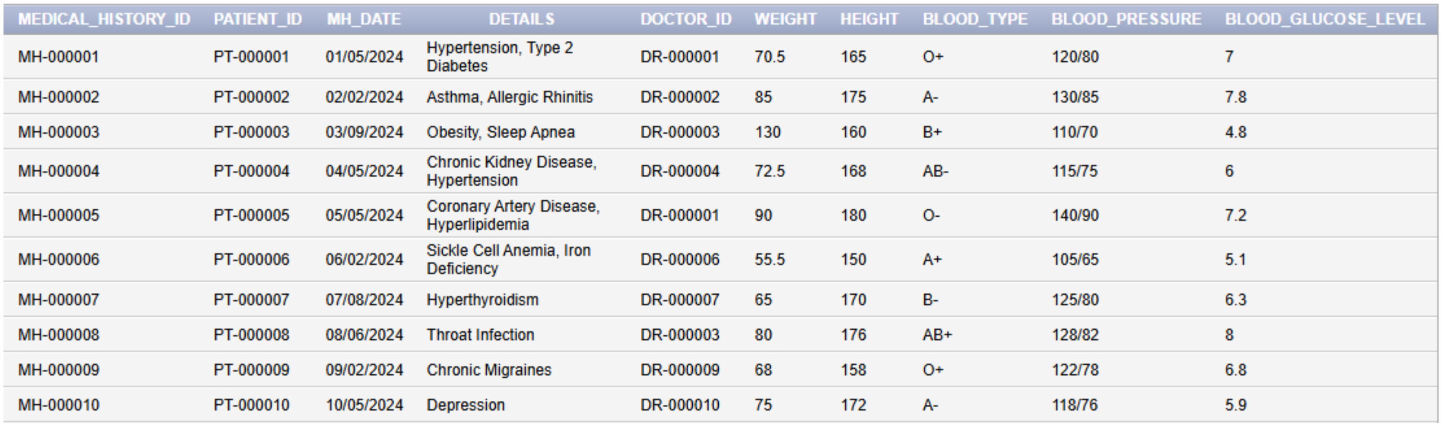
PHARMACY_ID	PHARMACY_NAME	PHONE_NUMBER	ADDRESS	OPERATING_HOURS
PH-001	Healthy Pharmacy	123-456-7890	123 Wellness St.	9 AM - 6 PM
PH-002	City Care Pharmacy	234-567-8901	456 City Ave.	8 AM - 8 PM
PH-003	CarePlus Pharmacy	345-678-9012	789 Care Rd.	10 AM - 5 PM
PH-004	Family Pharmacy	456-789-0123	101 Family Blvd.	9 AM - 7 PM
PH-005	Greenleaf Pharmacy	567-890-1234	202 Greenleaf Ln.	8 AM - 9 PM
PH-006	Sunshine Pharmacy	678-901-2345	303 Sunshine St.	9 AM - 6 PM
PH-007	CureWell Pharmacy	789-012-3456	404 Cure Rd.	8 AM - 7 PM
PH-008	PrimeCare Pharmacy	890-123-4567	505 Prime Blvd.	10 AM - 8 PM
PH-009	QuickMed Pharmacy	901-234-5678	606 Quick Lane	7 AM - 9 PM
PH-010	MedExpress Pharmacy	012-345-6789	707 Med Ave.	9 AM - 5 PM

10. Allergy



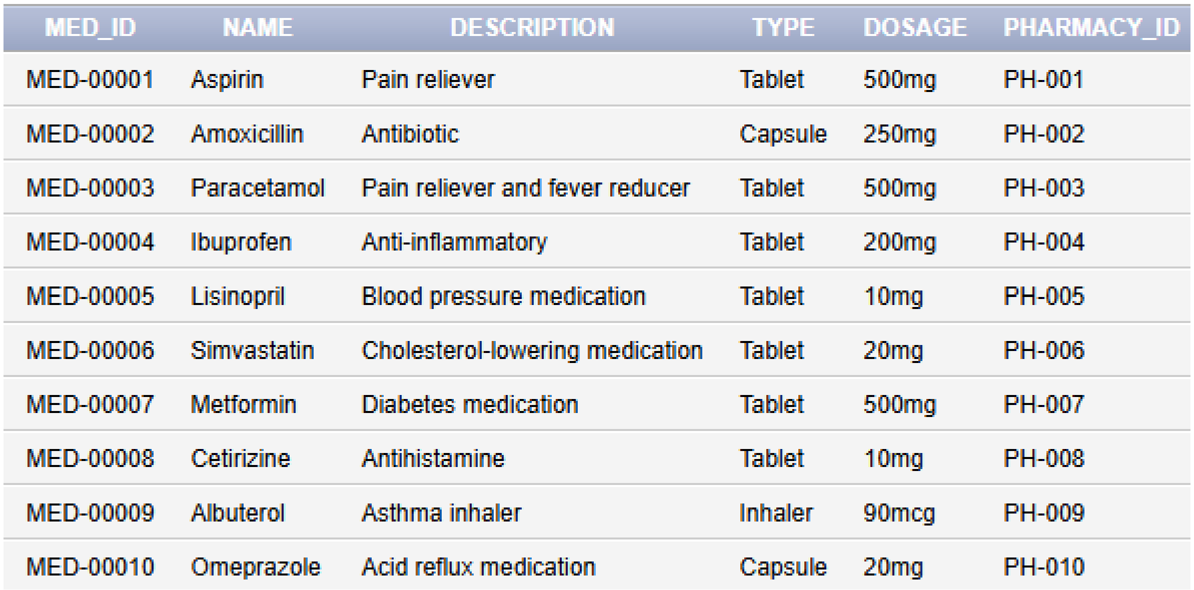
ALLERGY_ID	ALLERGEN_NAME	DESCRIPTION	SEVERITY	SYMPTOMS	PATIENT_ID
AL-100001	Shellfish	Food allergy	Severe	Hives, difficulty breathing	PT-000001
AL-100002	Tree Nuts	Food allergy	Severe	Swelling, anaphylaxis	PT-000002
AL-100003	Gluten	Food allergy	Moderate	Abdominal pain, bloating	PT-000003
AL-100004	Dairy	Food allergy	Moderate	Stomach cramps, diarrhea	PT-000004
AL-100005	Aspirin	Drug allergy	Severe	Rash, anaphylaxis	PT-000005
AL-100006	Amoxicillin	Drug allergy	High	Hives, fever	PT-000006
AL-100007	Ibuprofen	Drug allergy	Moderate	Stomach upset, rash	PT-000007
AL-100008	Grass	Environmental allergy	Moderate	Sneezing, itchy eyes	PT-000008
AL-100009	Birch Pollen	Environmental allergy	High	Sneezing, runny nose	PT-000009
AL-100010	Bees	Environmental allergy	Severe	Anaphylaxis, swelling	PT-000010
AL-100011	Latex	Environmental allergy	Severe	Skin rash, difficulty breathing	PT-000011

11. Medical_History



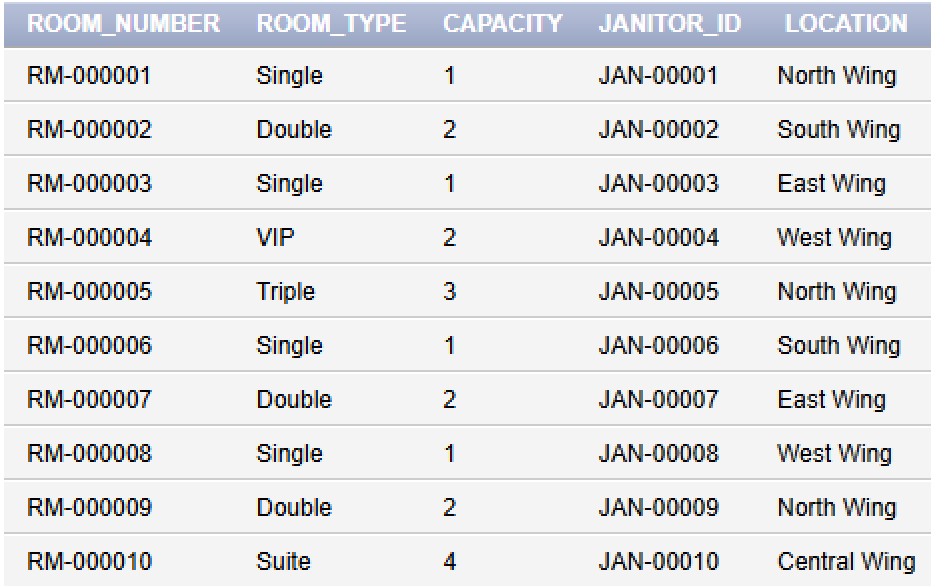
MEDICAL_HISTORY_ID	PATIENT_ID	MH_DATE	DETAILS	DOCTOR_ID	WEIGHT	HEIGHT	BLOOD_TYPE	BLOOD_PRESSURE	BLOOD_GLUCOSE_LEVEL
MH-000001	PT-000001	01/05/2024	Hypertension, Type 2 Diabetes	DR-000001	70.5	165	O+	120/80	7
MH-000002	PT-000002	02/02/2024	Asthma, Allergic Rhinitis	DR-000002	85	175	A-	130/85	7.8
MH-000003	PT-000003	03/09/2024	Obesity, Sleep Apnea	DR-000003	130	160	B+	110/70	4.8
MH-000004	PT-000004	04/05/2024	Chronic Kidney Disease, Hypertension	DR-000004	72.5	168	AB-	115/75	6
MH-000005	PT-000005	05/05/2024	Coronary Artery Disease, Hyperlipidemia	DR-000001	90	180	O-	140/90	7.2
MH-000006	PT-000006	06/02/2024	Sickle Cell Anemia, Iron Deficiency	DR-000006	55.5	150	A+	105/65	5.1
MH-000007	PT-000007	07/08/2024	Hyperthyroidism	DR-000007	65	170	B-	125/80	6.3
MH-000008	PT-000008	08/06/2024	Throat Infection	DR-000003	80	176	AB+	128/82	8
MH-000009	PT-000009	09/02/2024	Chronic Migraines	DR-000009	68	158	O+	122/78	6.8
MH-000010	PT-000010	10/05/2024	Depression	DR-000010	75	172	A-	118/76	5.9

12. Medication



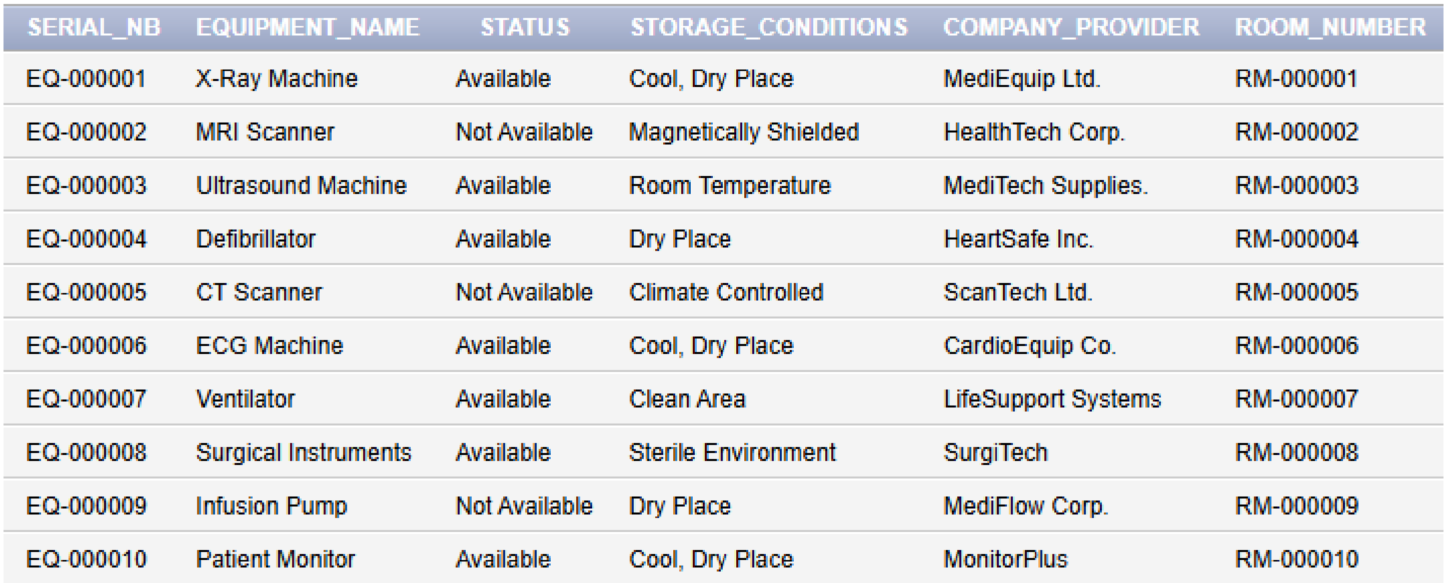
MED_ID	NAME	DESCRIPTION	TYPE	DOSAGE	PHARMACY_ID
MED-00001	Aspirin	Pain reliever	Tablet	500mg	PH-001
MED-00002	Amoxicillin	Antibiotic	Capsule	250mg	PH-002
MED-00003	Paracetamol	Pain reliever and fever reducer	Tablet	500mg	PH-003
MED-00004	Ibuprofen	Anti-inflammatory	Tablet	200mg	PH-004
MED-00005	Lisinopril	Blood pressure medication	Tablet	10mg	PH-005
MED-00006	Simvastatin	Cholesterol-lowering medication	Tablet	20mg	PH-006
MED-00007	Metformin	Diabetes medication	Tablet	500mg	PH-007
MED-00008	Cetirizine	Antihistamine	Tablet	10mg	PH-008
MED-00009	Albuterol	Asthma inhaler	Inhaler	90mcg	PH-009
MED-00010	Omeprazole	Acid reflux medication	Capsule	20mg	PH-010

13. Room



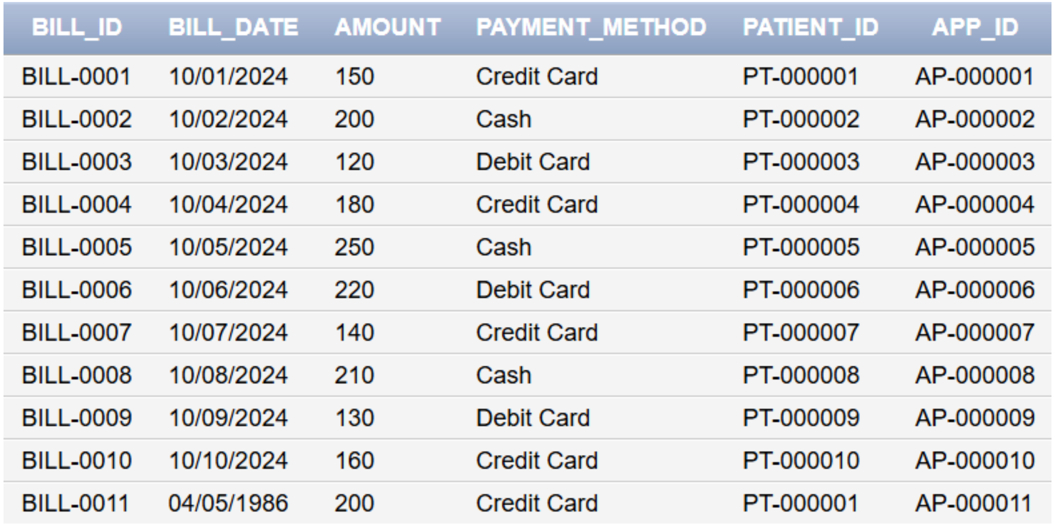
ROOM_NUMBER	ROOM_TYPE	CAPACITY	JANITOR_ID	LOCATION
RM-000001	Single	1	JAN-00001	North Wing
RM-000002	Double	2	JAN-00002	South Wing
RM-000003	Single	1	JAN-00003	East Wing
RM-000004	VIP	2	JAN-00004	West Wing
RM-000005	Triple	3	JAN-00005	North Wing
RM-000006	Single	1	JAN-00006	South Wing
RM-000007	Double	2	JAN-00007	East Wing
RM-000008	Single	1	JAN-00008	West Wing
RM-000009	Double	2	JAN-00009	North Wing
RM-000010	Suite	4	JAN-00010	Central Wing

14. Medical_Equipment



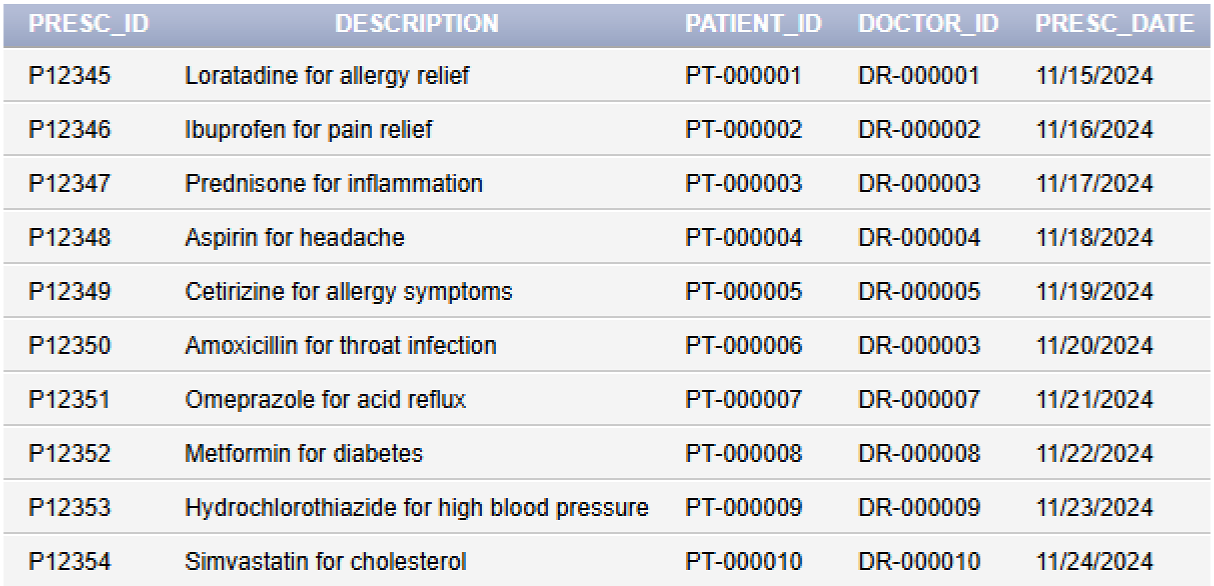
SERIAL_NB	EQUIPMENT_NAME	STATUS	STORAGE_CONDITIONS	COMPANY_PROVIDER	ROOM_NUMBER
EQ-000001	X-Ray Machine	Available	Cool, Dry Place	MediEquip Ltd.	RM-000001
EQ-000002	MRI Scanner	Not Available	Magnetically Shielded	HealthTech Corp.	RM-000002
EQ-000003	Ultrasound Machine	Available	Room Temperature	MediTech Supplies.	RM-000003
EQ-000004	Defibrillator	Available	Dry Place	HeartSafe Inc.	RM-000004
EQ-000005	CT Scanner	Not Available	Climate Controlled	ScanTech Ltd.	RM-000005
EQ-000006	ECG Machine	Available	Cool, Dry Place	CardioEquip Co.	RM-000006
EQ-000007	Ventilator	Available	Clean Area	LifeSupport Systems	RM-000007
EQ-000008	Surgical Instruments	Available	Sterile Environment	SurgiTech	RM-000008
EQ-000009	Infusion Pump	Not Available	Dry Place	MediFlow Corp.	RM-000009
EQ-000010	Patient Monitor	Available	Cool, Dry Place	MonitorPlus	RM-000010

15. Billing



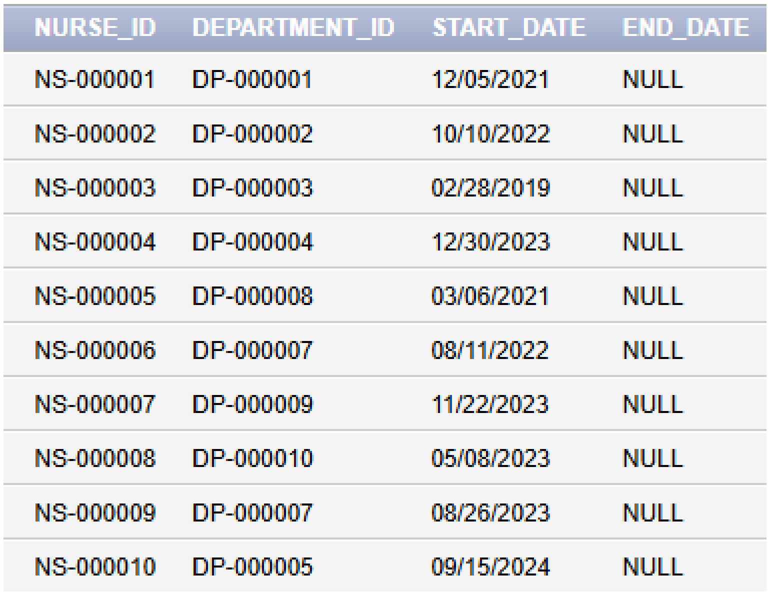
BILL_ID	BILL_DATE	AMOUNT	PAYMENT_METHOD	PATIENT_ID	APP_ID
BILL-0001	10/01/2024	150	Credit Card	PT-000001	AP-000001
BILL-0002	10/02/2024	200	Cash	PT-000002	AP-000002
BILL-0003	10/03/2024	120	Debit Card	PT-000003	AP-000003
BILL-0004	10/04/2024	180	Credit Card	PT-000004	AP-000004
BILL-0005	10/05/2024	250	Cash	PT-000005	AP-000005
BILL-0006	10/06/2024	220	Debit Card	PT-000006	AP-000006
BILL-0007	10/07/2024	140	Credit Card	PT-000007	AP-000007
BILL-0008	10/08/2024	210	Cash	PT-000008	AP-000008
BILL-0009	10/09/2024	130	Debit Card	PT-000009	AP-000009
BILL-0010	10/10/2024	160	Credit Card	PT-000010	AP-000010
BILL-0011	04/05/1986	200	Credit Card	PT-000001	AP-000011

16. Prescription



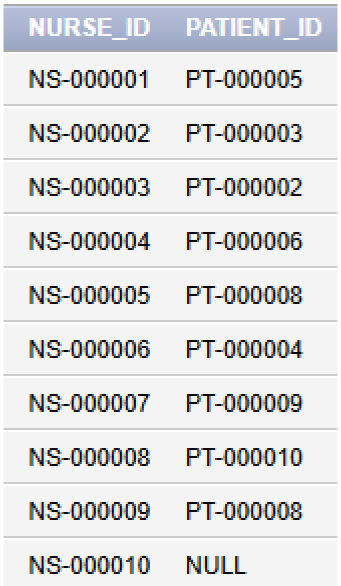
PRESC_ID	DESCRIPTION	PATIENT_ID	DOCTOR_ID	PRESC_DATE
P12345	Loratadine for allergy relief	PT-000001	DR-000001	11/15/2024
P12346	Ibuprofen for pain relief	PT-000002	DR-000002	11/16/2024
P12347	Prednisone for inflammation	PT-000003	DR-000003	11/17/2024
P12348	Aspirin for headache	PT-000004	DR-000004	11/18/2024
P12349	Cetirizine for allergy symptoms	PT-000005	DR-000005	11/19/2024
P12350	Amoxicillin for throat infection	PT-000006	DR-000003	11/20/2024
P12351	Omeprazole for acid reflux	PT-000007	DR-000007	11/21/2024
P12352	Metformin for diabetes	PT-000008	DR-000008	11/22/2024
P12353	Hydrochlorothiazide for high blood pressure	PT-000009	DR-000009	11/23/2024
P12354	Simvastatin for cholesterol	PT-000010	DR-000010	11/24/2024

17. Belongs_To2



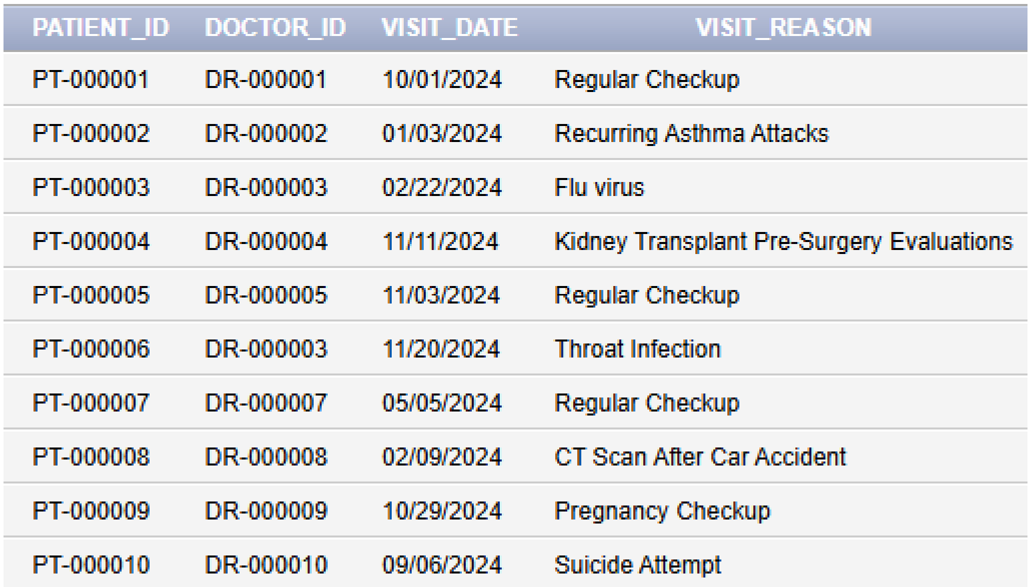
NURSE_ID	DEPARTMENT_ID	START_DATE	END_DATE
NS-000001	DP-000001	12/05/2021	NULL
NS-000002	DP-000002	10/10/2022	NULL
NS-000003	DP-000003	02/28/2019	NULL
NS-000004	DP-000004	12/30/2023	NULL
NS-000005	DP-000008	03/06/2021	NULL
NS-000006	DP-000007	08/11/2022	NULL
NS-000007	DP-000009	11/22/2023	NULL
NS-000008	DP-000010	05/08/2023	NULL
NS-000009	DP-000007	08/26/2023	NULL
NS-000010	DP-000005	09/15/2024	NULL

18. Services



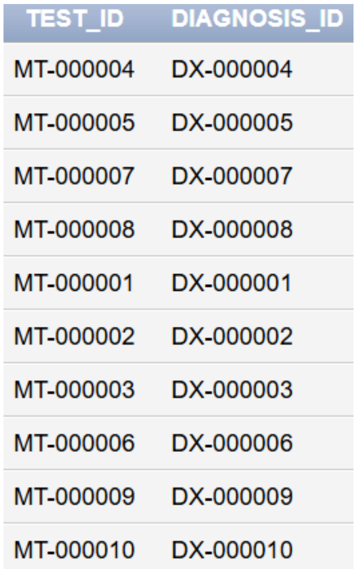
NURSE_ID	PATIENT_ID
NS-000001	PT-000005
NS-000002	PT-000003
NS-000003	PT-000002
NS-000004	PT-000006
NS-000005	PT-000008
NS-000006	PT-000004
NS-000007	PT-000009
NS-000008	PT-000010
NS-000009	PT-000008
NS-000010	NULL

19. Visits



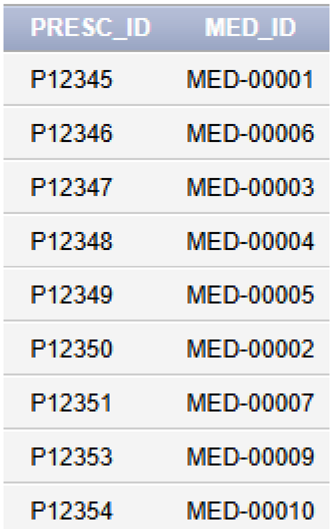
PATIENT_ID	DOCTOR_ID	VISIT_DATE	VISIT_REASON
PT-000001	DR-000001	10/01/2024	Regular Checkup
PT-000002	DR-000002	01/03/2024	Recurring Asthma Attacks
PT-000003	DR-000003	02/22/2024	Flu virus
PT-000004	DR-000004	11/11/2024	Kidney Transplant Pre-Surgery Evaluations
PT-000005	DR-000005	11/03/2024	Regular Checkup
PT-000006	DR-000003	11/20/2024	Throat Infection
PT-000007	DR-000007	05/05/2024	Regular Checkup
PT-000008	DR-000008	02/09/2024	CT Scan After Car Accident
PT-000009	DR-000009	10/29/2024	Pregnancy Checkup
PT-000010	DR-000010	09/06/2024	Suicide Attempt

20. Requires



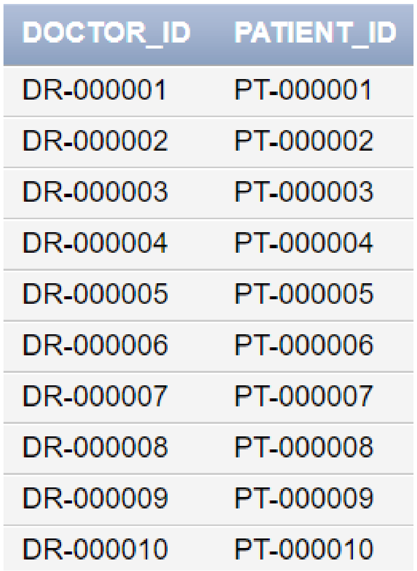
TEST_ID	DIAGNOSIS_ID
MT-000004	DX-000004
MT-000005	DX-000005
MT-000007	DX-000007
MT-000008	DX-000008
MT-000001	DX-000001
MT-000002	DX-000002
MT-000003	DX-000003
MT-000006	DX-000006
MT-000009	DX-000009
MT-000010	DX-000010

21. Contains1



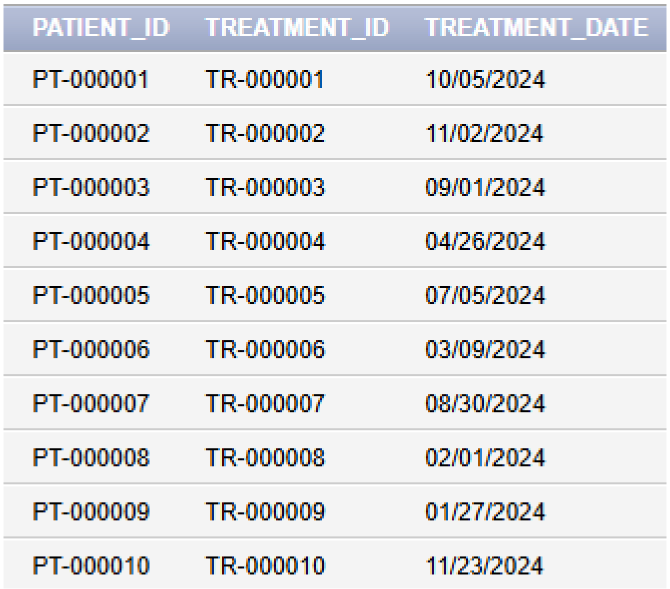
PRESC_ID	MED_ID
P12345	MED-00001
P12346	MED-00006
P12347	MED-00003
P12348	MED-00004
P12349	MED-00005
P12350	MED-00002
P12351	MED-00007
P12353	MED-00009
P12354	MED-00010

22. Treats



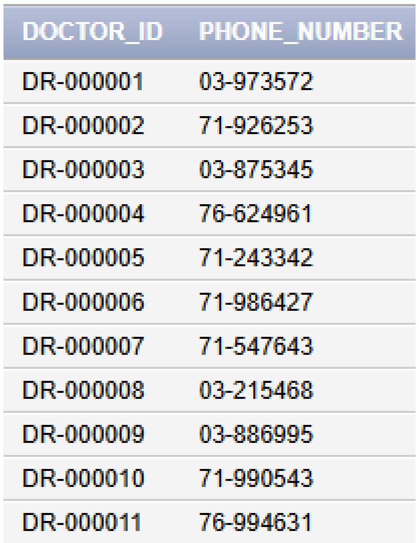
DOCTOR_ID	PATIENT_ID
DR-000001	PT-000001
DR-000002	PT-000002
DR-000003	PT-000003
DR-000004	PT-000004
DR-000005	PT-000005
DR-000006	PT-000006
DR-000007	PT-000007
DR-000008	PT-000008
DR-000009	PT-000009
DR-000010	PT-000010

23. Undergoes



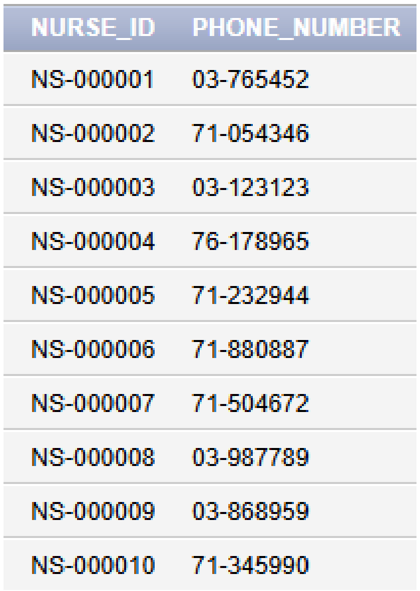
PATIENT_ID	TREATMENT_ID	TREATMENT_DATE
PT-000001	TR-000001	10/05/2024
PT-000002	TR-000002	11/02/2024
PT-000003	TR-000003	09/01/2024
PT-000004	TR-000004	04/26/2024
PT-000005	TR-000005	07/05/2024
PT-000006	TR-000006	03/09/2024
PT-000007	TR-000007	08/30/2024
PT-000008	TR-000008	02/01/2024
PT-000009	TR-000009	01/27/2024
PT-000010	TR-000010	11/23/2024

24. Doctor_Phone_Number



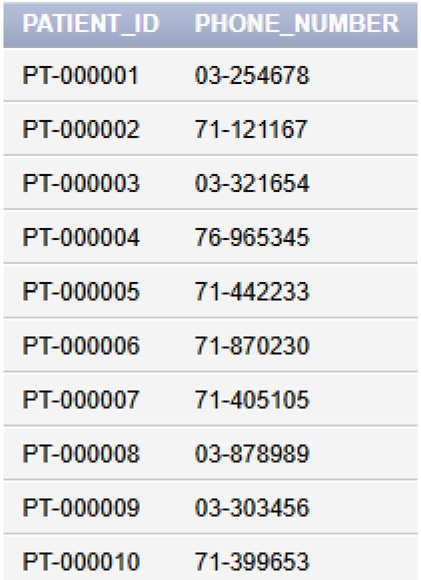
DOCTOR_ID	PHONE_NUMBER
DR-000001	03-973572
DR-000002	71-926253
DR-000003	03-875345
DR-000004	76-624961
DR-000005	71-243342
DR-000006	71-986427
DR-000007	71-547643
DR-000008	03-215468
DR-000009	03-886995
DR-000010	71-990543
DR-000011	76-994631

25. Nurse_Phone_Number



NURSE_ID	PHONE_NUMBER
NS-000001	03-765452
NS-000002	71-054346
NS-000003	03-123123
NS-000004	76-178965
NS-000005	71-232944
NS-000006	71-880887
NS-000007	71-504672
NS-000008	03-987789
NS-000009	03-868959
NS-000010	71-345990

26. Patient_Phone_Number



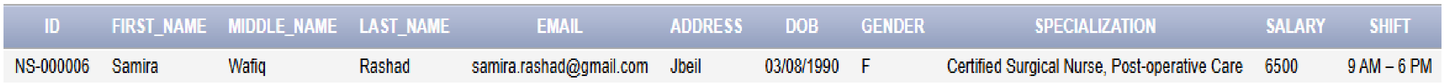
PATIENT_ID	PHONE_NUMBER
PT-000001	03-254678
PT-000002	71-121167
PT-000003	03-321654
PT-000004	76-965345
PT-000005	71-442233
PT-000006	71-870230
PT-000007	71-405105
PT-000008	03-878989
PT-000009	03-303456
PT-000010	71-399653

XIV. Queries:

Query 1: The Case of the Hopeless Romantic Patient

Patient x is a hopeless romantic who underwent surgery in our clinic. After his surgery, Nurse Samira was assigned to take care of his stay. He couldn't help but admire the compassion and care Nurse Samira showed him during his recovery. As the hopeless romantic he is, he fell in love with her. Unfortunately, luck was not on his side. When he wanted to confess his feelings on his last day, she was not on duty. Determined to find her and confess, he asked his hacker friend to help him to get into the database to find her information. All he remembered was that she worked in the surgery department and she serviced him after the surgery.

```sql
SELECT N.*
FROM NURSE N
JOIN BELONGS_TO2 B ON N.ID = B.NURSE_ID
JOIN DEPARTMENT D ON B.DEPARTMENT_ID = D.DEPARTMENT_ID
JOIN SERVICES S ON N.ID = S.NURSE_ID
WHERE S.PATIENT_ID = 'PT-000004'
      AND D.DEPARTMENT_NAME = 'Surgery';
```

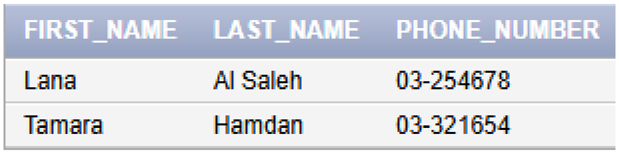


ID	FIRST_NAME	MIDDLE_NAME	LAST_NAME	EMAIL	ADDRESS	DOB	GENDER	SPECIALIZATION	SALARY	SHIFT
NS-000006	Samira	Wafiq	Rashad	samira.rashad@gmail.com	Jbeil	03/08/1990	F	Certified Surgical Nurse, Post-operative Care	6500	9 AM – 6 PM

Query 2: The Case of the Missing Wallet

Doctor Hassan Itani was panicking because could not find his wallet anywhere. He searched his house as well as his office in the clinic and it was nowhere to be seen. He vividly recalled having it during his appointments on November 1. The possibility of a patient accidentally picking it up was gnawing at him. Determined to retrieve it, Doctor Hassan decided to check the records of the patients who had appointments with him on that day to contact them and solve the mystery.

```sql
SELECT P.FIRST_NAME, P.LAST_NAME, PP.PHONE_NUMBER
FROM APPOINTMENT A
JOIN PATIENT P ON A.PATIENT_ID = P.ID
JOIN PATIENT_PHONE_NUMBER PP ON P.ID = PP.PATIENT_ID
WHERE A.DOCTOR_ID = 'DR-000001'
      AND A.APP_DATE = TO_DATE('11-01-2024', 'MM-DD-YYYY');
```

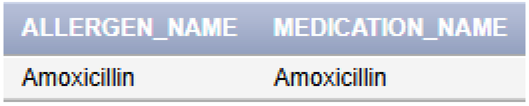


FIRST_NAME	LAST_NAME	PHONE_NUMBER
Lana	Al Saleh	03-254678
Tamara	Hamdan	03-321654

Query 3: The Case of the Vengeful Doctor

Patient Khaled Daaboul stormed into the hospital, claiming that Doctor Sally prescribed him a medication containing a substance he is allergic to when he visited him with an infected throat. The patient was furious and wanted to sue the doctor. The doctor claims that she is innocent and did not know that the patient had this allergy. The clinic needed to check whether the patient had an allergic reaction due to the doctor's prescription.

```
SELECT AL.ALLERGEN_NAME AS ALLERGEN_NAME, M.NAME AS MEDICATION_NAME
FROM ALLERGY AL
JOIN PATIENT P ON AL.PATIENT_ID = P.ID
JOIN PRESCRIPTION PR ON P.ID = PR.PATIENT_ID
JOIN CONTAINS1 C ON PR.PRESC_ID = C.PRESC_ID
JOIN MEDICATION M ON C.MED_ID = M.MED_ID
WHERE P.ID = 'PT-000006'
      AND PR.DOCTOR_ID = 'DR-000003'
      AND (AL.ALLERGEN_NAME LIKE '%Amoxicillin%' OR M.NAME LIKE '%Amoxicillin%');
```



ALLERGEN_NAME	MEDICATION_NAME
Amoxicillin	Amoxicillin

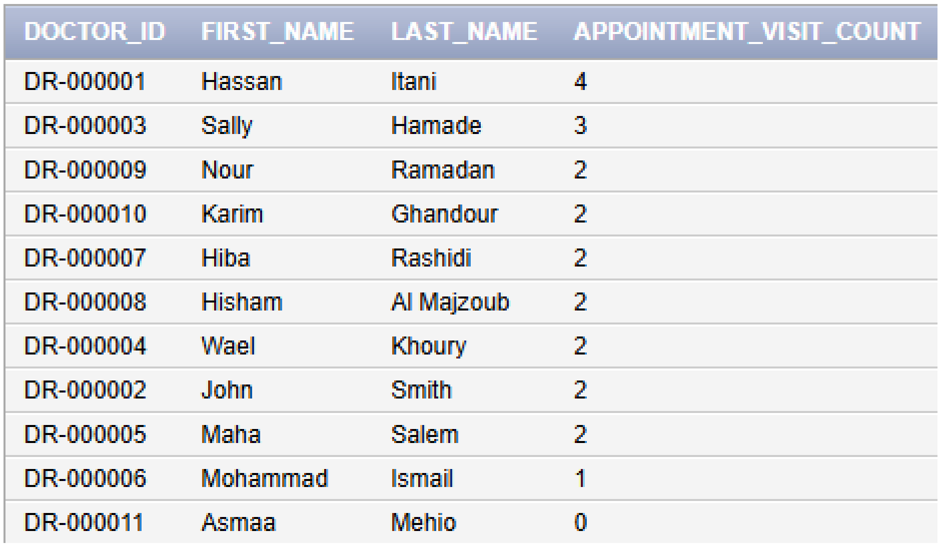
After the police's investigation, it turned out that Doctor Sally wanted to take revenge on the patient, since he previously criticized her medical decisions. Fueled by anger, Doctor Sally intentionally prescribed the medication that contained an allergen the patient was sensitive to. Thus, the clinic immediately fired Doctor Sally and she was sued for her wrongdoings. The appointments that were scheduled for Doctor Sally after 12-01-2024 will be assigned to Doctor Asmaa an internal medicine doctor in our clinic.

```
UPDATE APPOINTMENT
SET DOCTOR_ID = 'DR-000011'
WHERE DOCTOR_ID = 'DR-000003'
AND APP_DATE > TO_DATE('12-01-2024', 'MM-DD-YYYY');
```

Query 4: The Case of the Dedicated Doctors

Dr. Hassan Itani, a cardiologist, was known for his tireless dedication to his patients at the clinic. He had treated many patients with heart conditions over the years, making his name synonymous with care and expertise. However, his colleague, Dr. Wael Khoury, an immunologist, was also gaining a reputation for handling complex cases with precision. The owner of the clinic wanted to know how much each doctor is contributing towards the clinic. Thus, this code will be used to help him in knowing the number of appointments and visits for each doctor. Rewarding them later for their hard work.

```sql
SELECT D.ID AS DOCTOR_ID,
       D.FIRST_NAME,
       D.LAST_NAME,
       COUNT(DISTINCT A.APPOINTMENT_NB) + COUNT(DISTINCT V.PATIENT_ID) AS
APPOINTMENT_VISIT_COUNT
FROM DOCTOR D
LEFT JOIN APPOINTMENT A ON D.ID = A.DOCTOR_ID
LEFT JOIN VISITS V ON D.ID = V.DOCTOR_ID
GROUP BY D.ID, D.FIRST_NAME, D.LAST_NAME
ORDER BY APPOINTMENT_VISIT_COUNT DESC;
```

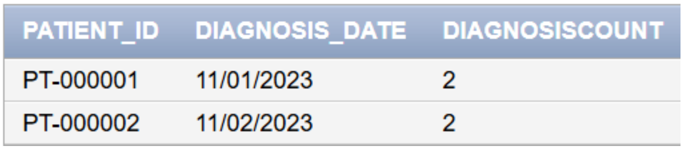


DOCTOR_ID	FIRST_NAME	LAST_NAME	APPOINTMENT_VISIT_COUNT
DR-000001	Hassan	Itani	4
DR-000003	Sally	Hamade	3
DR-000009	Nour	Ramadan	2
DR-000010	Karim	Ghandour	2
DR-000007	Hiba	Rashidi	2
DR-000008	Hisham	Al Majzoub	2
DR-000004	Wael	Khoury	2
DR-000002	John	Smith	2
DR-000005	Maha	Salem	2
DR-000006	Mohammad	Ismail	1
DR-000011	Asmaa	Mehio	0

Query 5: The Case of Duplicate Identification Diagnosis

In the healthcare context, it's important to ensure that medical records are accurate and not duplicated. A situation can arise where due to a system error, a patient may get diagnosed multiple times on the same day for the same or different conditions. This is problematic because it could lead to incorrect billing, reporting errors, and unnecessary treatment planning. In this case, the objective is to identify patients who have been diagnosed multiple times on the same day so that the records can be verified and corrected if necessary.

```sql
SELECT PATIENT_ID, DIAGNOSIS_DATE, COUNT(*) AS DIAGNOSISCOUNT
FROM DIAGNOSIS
GROUP BY PATIENT_ID, DIAGNOSIS_DATE
HAVING COUNT(*) > 1;
```

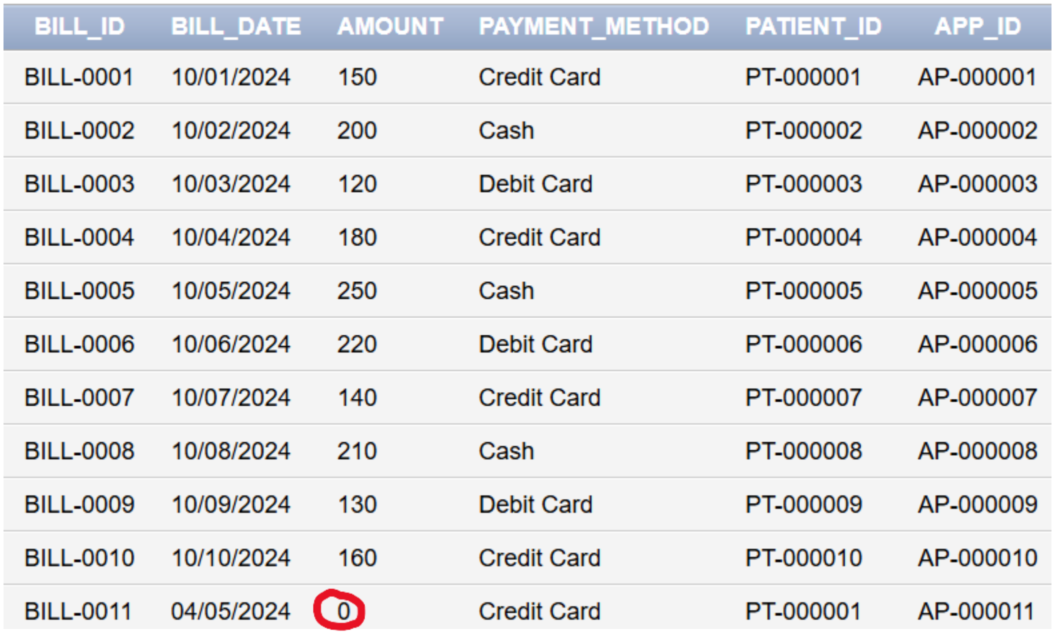


PATIENT_ID	DIAGNOSIS_DATE	DIAGNOSISCOUNT
PT-000001	11/01/2023	2
PT-000002	11/02/2023	2

Query 6: Identifying Patients with Birthday Discount or Free Service

This query is designed to check if a patient's birthday falls on the same day as their appointment. If this is the case, the patient may be eligible for a special discount or even a free consultation. The query checks the patient's date of birth (DOB) and compares it to the appointment date. If the two dates match, the system could flag the patient for a potential benefit, like a birthday discount.

```
UPDATE BILLING B

SET B.AMOUNT = 0

WHERE EXISTS (

    SELECT 1

    FROM APPOINTMENT A

    JOIN PATIENT P ON A.PATIENT_ID = P.ID

    WHERE B.PATIENT_ID = P.ID

    AND B.APP_ID = A.APPOINTMENT_NB

    AND TO_CHAR(P.DOB, 'MM-DD') = TO_CHAR(A.APP_DATE, 'MM-DD')

);
```



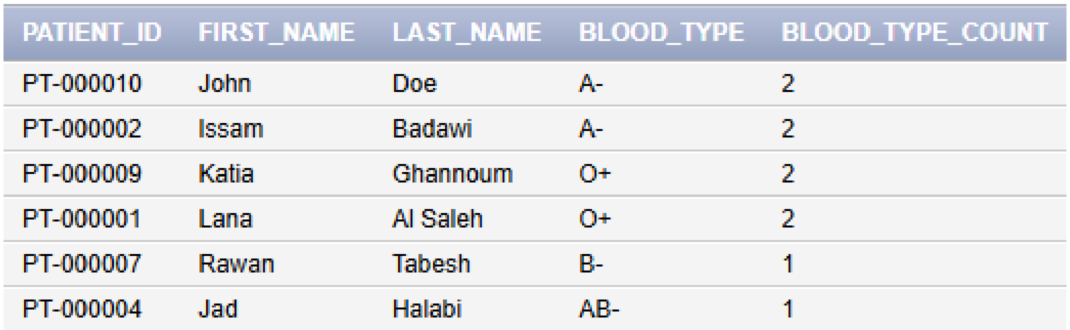
BILL_ID	BILL_DATE	AMOUNT	PAYMENT_METHOD	PATIENT_ID	APP_ID
BILL-0001	10/01/2024	150	Credit Card	PT-000001	AP-000001
BILL-0002	10/02/2024	200	Cash	PT-000002	AP-000002
BILL-0003	10/03/2024	120	Debit Card	PT-000003	AP-000003
BILL-0004	10/04/2024	180	Credit Card	PT-000004	AP-000004
BILL-0005	10/05/2024	250	Cash	PT-000005	AP-000005
BILL-0006	10/06/2024	220	Debit Card	PT-000006	AP-000006
BILL-0007	10/07/2024	140	Credit Card	PT-000007	AP-000007
BILL-0008	10/08/2024	210	Cash	PT-000008	AP-000008
BILL-0009	10/09/2024	130	Debit Card	PT-000009	AP-000009
BILL-0010	10/10/2024	160	Credit Card	PT-000010	AP-000010
BILL-0011	04/05/2024	0	Credit Card	PT-000001	AP-000011

This sample output shows that Lana Moustafa has an appointment on her birthday and would be eligible for a birthday discount or free service.

Query 7: The Emergency Blood Rush

Dr. John Smith received an urgent call about a critical shortage of certain blood types in the hospital. To ensure that the hospital was fully prepared for any emergencies, he needed to identify patients with blood types that were in high demand, particularly the rare ones. Dr. Smith quickly reviewed the medical records to identify patients with blood types such as **O+, A-, B-, and AB-**, as these were the blood types in shortest supply and would be prioritized in case of an emergency.

```
SELECT PATIENT.ID AS PATIENT_ID,PATIENT.FIRST_NAME,PATIENT.LAST_NAME,

MEDICAL_HISTORY.BLOOD_TYPE,BLOOD_TYPE_COUNT.BLOOD_TYPE_COUNT

FROM PATIENT

JOIN MEDICAL_HISTORY ON PATIENT.ID = MEDICAL_HISTORY.PATIENT_ID

JOIN (SELECT MEDICAL_HISTORY.BLOOD_TYPE,COUNT(PATIENT.ID) AS BLOOD_TYPE_COUNT

      FROM PATIENT

      JOIN MEDICAL_HISTORY ON PATIENT.ID = MEDICAL_HISTORY.PATIENT_ID

      WHERE MEDICAL_HISTORY.BLOOD_TYPE IN ('O+','A-','B-','AB-')

      GROUP BY MEDICAL_HISTORY.BLOOD_TYPE) BLOOD_TYPE_COUNT

ON MEDICAL_HISTORY.BLOOD_TYPE = BLOOD_TYPE_COUNT.BLOOD_TYPE

WHERE MEDICAL_HISTORY.BLOOD_TYPE IN ('O+','A-','B-','AB-');
```



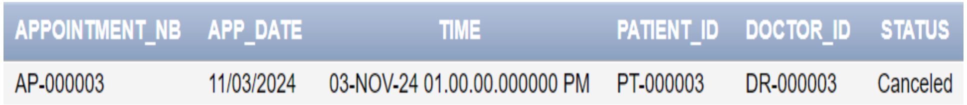
PATIENT_ID	FIRST_NAME	LAST_NAME	BLOOD_TYPE	BLOOD_TYPE_COUNT
PT-000010	John	Doe	A-	2
PT-000002	Issam	Badawi	A-	2
PT-000009	Katia	Ghannoum	O+	2
PT-000001	Lana	Al Saleh	O+	2
PT-000007	Rawan	Tabesh	B-	1
PT-000004	Jad	Halabi	AB-	1

After reviewing the records, Dr. Smith identified several patients who required priority attention due to their rare blood types. He instructed the hospital staff to take necessary precautions, ensuring these patients would receive immediate care if an emergency arose.

Query 8: The Case of Canceling An Appointment

Tamara Hamdan, a patient scheduled for an appointment with Dr. John Smith on November 3, 2024, decided to cancel her visit. To update the doctor's schedule and free up the time slot, the hospital staff did not delete the appointment from the system. Instead, they updated the appointment's status to "Canceled." The system then showed only the canceled appointments, ensuring that the time slot was marked as unavailable and could be reviewed for rescheduling or reallocation.

```
ALTER TABLE APPOINTMENT

ADD STATUS VARCHAR(20);

- - - - - - - - - - -

UPDATE APPOINTMENT

SET STATUS = 'Canceled'

WHERE APPOINTMENT_NB = 'AP-000003';

- - - - - - - - - - -

SELECT APPOINTMENT_NB, APP_DATE, TIME, PATIENT_ID, DOCTOR_ID, STATUS

FROM APPOINTMENT

WHERE STATUS = 'Canceled';
```

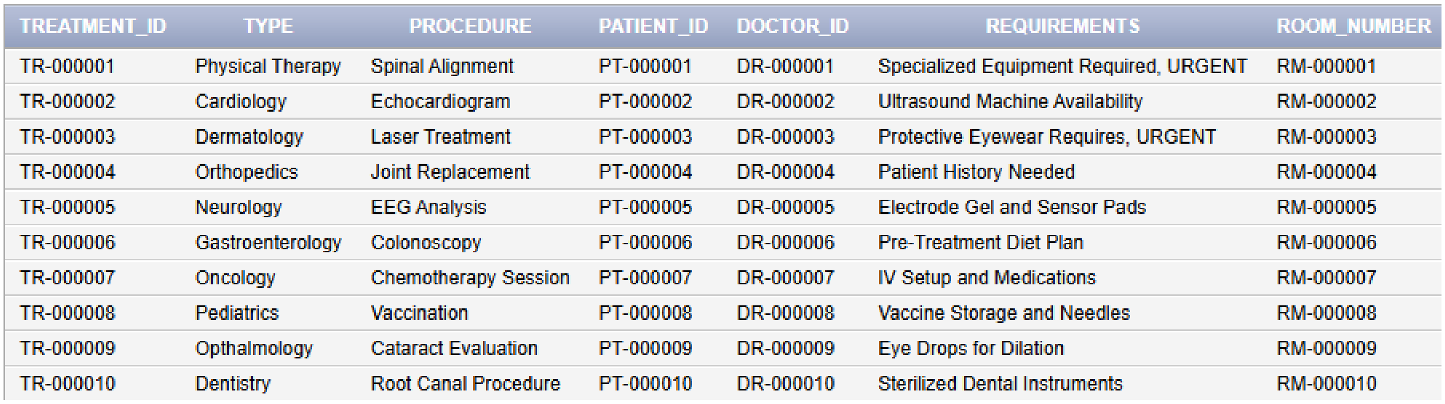


APPOINTMENT_NB	APP_DATE	TIME	PATIENT_ID	DOCTOR_ID	STATUS
AP-000003	11/03/2024	03-NOV-24 01.00.00.000000 PM	PT-000003	DR-000003	Canceled

Query 9: Prioritizing Patients Requiring Immediate Treatment

There are some patients who require immediate treatment. This SQL code will help the clinic in identifying the patients who need to receive immediate treatments as soon as possible. As a result, the clinic could do their treatments before their condition gets worse.

```
UPDATE TREATMENT T
SET T.REQUIREMENTS = CONCAT(T.REQUIREMENTS, ', URGENT')
WHERE EXISTS (
    SELECT *
    FROM DIAGNOSIS D
    JOIN PATIENT P ON D.PATIENT_ID = P.ID
    WHERE D.PATIENT_ID = T.PATIENT_ID
    AND D.DESCRIPTION LIKE '%Emergency%'
);
SELECT * FROM TREATMENT;
```

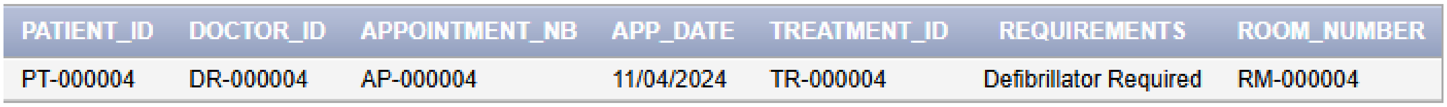


TREATMENT_ID	TYPE	PROCEDURE	PATIENT_ID	DOCTOR_ID	REQUIREMENTS	ROOM_NUMBER
TR-000001	Physical Therapy	Spinal Alignment	PT-000001	DR-000001	Specialized Equipment Required, URGENT	RM-000001
TR-000002	Cardiology	Echocardiogram	PT-000002	DR-000002	Ultrasound Machine Availability	RM-000002
TR-000003	Dermatology	Laser Treatment	PT-000003	DR-000003	Protective Eyewear Requires, URGENT	RM-000003
TR-000004	Orthopedics	Joint Replacement	PT-000004	DR-000004	Patient History Needed	RM-000004
TR-000005	Neurology	EEG Analysis	PT-000005	DR-000005	Electrode Gel and Sensor Pads	RM-000005
TR-000006	Gastroenterology	Colonoscopy	PT-000006	DR-000006	Pre-Treatment Diet Plan	RM-000006
TR-000007	Oncology	Chemotherapy Session	PT-000007	DR-000007	IV Setup and Medications	RM-000007
TR-000008	Pediatrics	Vaccination	PT-000008	DR-000008	Vaccine Storage and Needles	RM-000008
TR-000009	Opthalmology	Cataract Evaluation	PT-000009	DR-000009	Eye Drops for Dilation	RM-000009
TR-000010	Dentistry	Root Canal Procedure	PT-000010	DR-000010	Sterilized Dental Instruments	RM-000010

Query 10: The Case of the Missing Defibrillator

A defibrillator has gone missing from RM-000004 at the clinic. We need to track down which patients might have used that room on the day it went missing.

```
SELECT A.PATIENT_ID, A.DOCTOR_ID, A.APPOINTMENT_NB, A.APP_DATE, T.TREATMENT_ID,
T.REQUIREMENTS, T.ROOM_NUMBER

FROM APPOINTMENT A

JOIN TREATMENT T ON A.PATIENT_ID = T.PATIENT_ID AND A.DOCTOR_ID = T.DOCTOR_ID

WHERE A.APP_DATE > TO_DATE('2024-11-01', 'YYYY-MM-DD')

AND T.REQUIREMENTS LIKE '%Defibrillator%'

AND T.ROOM_NUMBER = 'RM-000004';
```



PATIENT_ID	DOCTOR_ID	APPOINTMENT_NB	APP_DATE	TREATMENT_ID	REQUIREMENTS	ROOM_NUMBER
PT-000004	DR-000004	AP-000004	11/04/2024	TR-000004	Defibrillator Required	RM-000004

XV. Normalization Up to The BCNF Normal Form:

After creating all relations we should improve them by normalizing according to several normal forms. Here we are going to normalize our database up to the fourth normal form which is the Boyce Codd Normal Form. On each relation we are going to apply the four normal forms. We start with the first then second then third and at last the BCNF normal form. Let us first start by a general description to each normal form.

First Normal Form:

This form does not allow multivalued attributes, composite attributes, and their combinations to exist in a relation.

1. Only attribute values permitted are single atomic values.
2. Domain of an attribute must only include atomic values and the value of an attribute in a tuple must be a single value from the domain of that attribute.
3. Disallows having a set of values as an attribute value for a single tuple.

Second Normal Form:

The Second normal form is based on the concept of full functional dependency. Before explaining the second form let us define some concepts used in this form and other forms also.

Functional Dependencies: A constraint between two sets of attributes from the database.

The values of the Y component of a tuple in relation R depend on, or are determined by the values of an X component.

We say that Y is functionally dependent on X.

Prime attribute: An attribute that is a member of a candidate key in a relation R.

An attribute is called non prime if it is not a prime attribute that is, if it is not a member of any candidate key.

Full functional dependency: A functional dependency X →Y is a full functional dependency if removal of any attribute A from X means that the dependency does not hold anymore.

Partial Dependency: A functional dependency X →Y is a partial functional dependency if removal of any attribute A from X means that the dependency still holds.

A relation schema R is in the second normal form if every nonprime attribute in R is fully functionally dependent on every key of R and every nonprime attribute A in R is not partially dependent on any key in R.

Third Normal Form:

The third normal form is based on the concept of transitive dependency. So let us first define a transitive dependency.

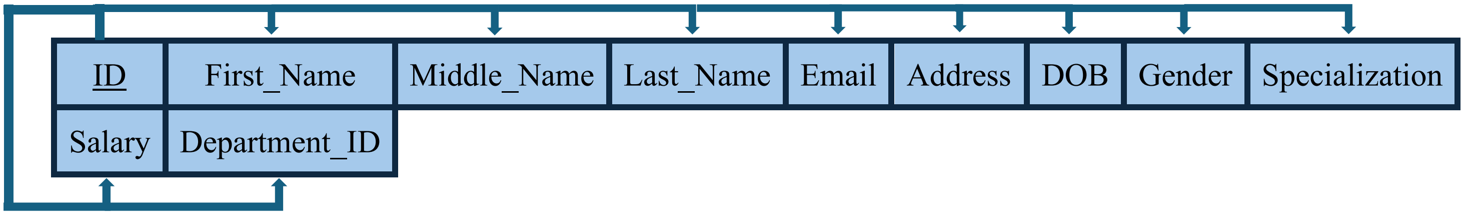
Transitive Dependency: A functional dependency X →Y in a relation schema R is a transitive dependency if there exists a set of attributes Z in R that is neither a candidate key nor a subset of any key of R, and both X →Z and Z →Y hold.

A relation schema R is in the third normal form if it satisfies the second normal form and no nonprime attribute of R is transitively dependent on the primary key. For every nontrivial functional dependency X →Y either X should be a super key or Y is a prime attribute.

Boycee Codd Normal Form:

The Boycee Codd normal form is a stricter form than the third normal form. The BCNF differs from the definition of the third normal form in only one condition. The third normal form allows the right hand side of the functional dependency to be a prime attribute while BCNF does not allow that.

1. Doctor



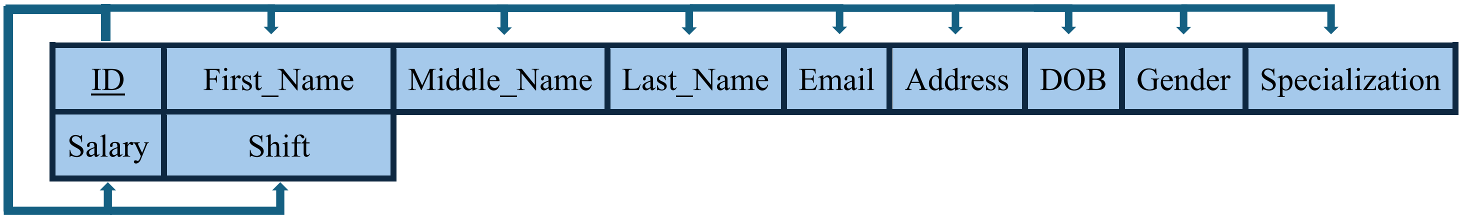
A. The **DOCTOR** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.

B. The **DOCTOR** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key "ID".

C. The **DOCTOR** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there is no non-prime attributes that are transitively dependent on the primary key "ID".

D. The **DOCTOR** relation schema satisfies all conditions of the BCNF because there exists no functional dependency X→A where X is not a super key or A is a prime attribute and X not a super key.

2. Nurse



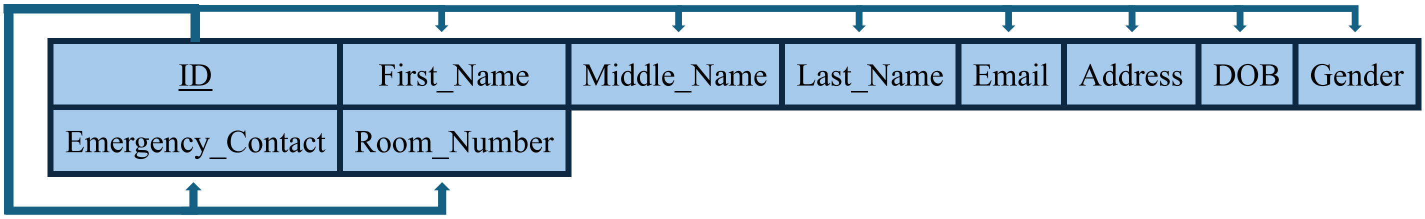
A. The **NURSE** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.

B. The **NURSE** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key "ID".

C. The **NURSE** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there is no non-prime attributes that are transitively dependent on the primary key "ID".

D. The **NURSE** relation schema satisfies all conditions of the BCNF because there exists no functional dependency X→A where X is not a super key or A is a prime attribute and X not a super key.

3. Patient



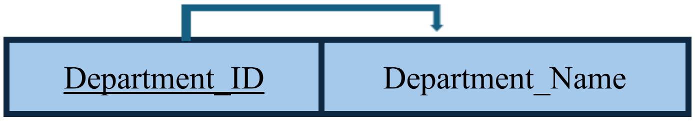
A. The **PATIENT** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.

B. The **PATIENT** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key "ID".

C. The **PATIENT** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there is no non-prime attributes that are transitively dependent on the primary key "ID".

D. The **PATIENT** relation schema satisfies all conditions of the BCNF because there exists no functional dependency X→A where X is not a super key or A is a prime attribute and X not a super key.

4. Department



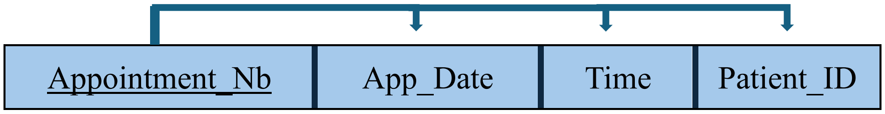
A. The **DEPARTMENT** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.

B. The **DEPARTMENT** relation schema satisfies all conditions of the 2NF because every nonprime attribute (Department_Name) is fully functionally dependent on the primary key "Department_ID".

C. The **DEPARTMENT** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF, and there is no non-prime attribute that is transitively dependent on the primary key "Department_ID".

D. The **DEPARTMENT** relation schema satisfies all conditions of the BCNF because there exists no functional dependency X → A where X is not a super key or A is a prime attribute or A is a prime attribute and X not a super key.

5. Appointment



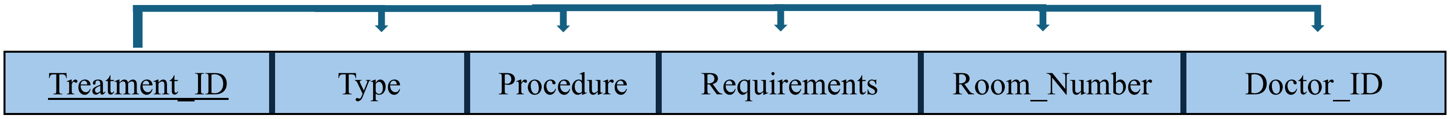
A. The **APPOINTMENT** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.

B. The **APPOINTMENT** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key “Appointment_Nb”.

C. The **APPOINTMENT** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there is no non-prime attributes that are transitively dependent on the primary key “Appointment_Nb”.

D. The **APPOINTMENT** relation schema satisfies all conditions of the BCNF because there exists no functional dependency X→A where X is not a super key or A is a prime attribute and X not a super key.

6. Treatment



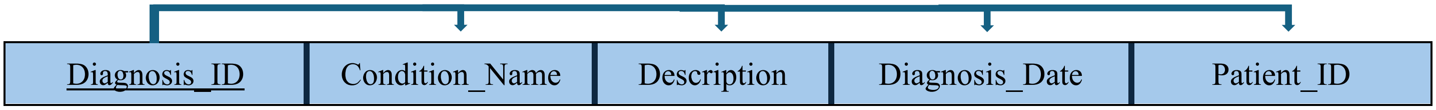
A. The **TREATMENT** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.

B. The **TREATMENT** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key “Treatment_ID”.

C. The **TREATMENT** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there is no non-prime attributes that are transitively dependent on the primary key “Treatment_ID”.

D. The **TREATMENT** relation schema satisfies all conditions of the BCNF because there exists no functional dependency X→A where X is not a super key or A is a prime attribute and X not a super key.

7. Diagnosis



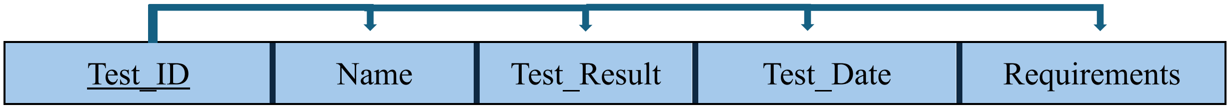
A. The **DIAGNOSIS** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.

B. The **DIAGNOSIS** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key “Diagnosis_ID”.

C. The **DIAGNOSIS** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there is no non-prime attributes that are transitively dependent on the primary key “Diagnosis_ID”.

D. The **DIAGNOSIS** relation schema satisfies all conditions of the BCNF because there exists no functional dependency X→A where X is not a super key or A is a prime attribute and X not a super key.

8. Medical_Test



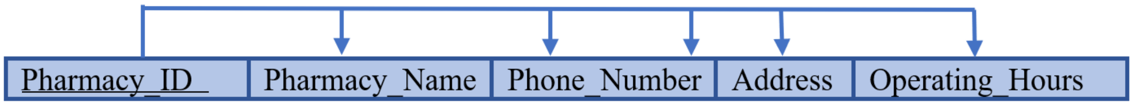
A. The **MEDICAL_TEST** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.

B. The **MEDICAL_TEST** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key “Test_ID”.

C. The **MEDICAL_TEST** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there is no non-prime attributes that are transitively dependent on the primary key “Test_ID”.

D. The **MEDICAL_TEST** relation schema satisfies all conditions of the BCNF because there exists no functional dependency X→A where X is not a super key or A is a prime attribute and X not a super key.

9. Pharmacy



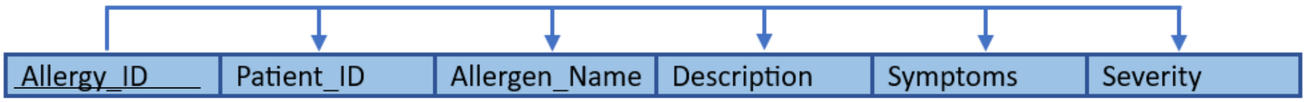
A. The **PHARMACY** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.

B. The **PHARMACY** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key "Pharmacy_ID".

C. The **PHARMACY** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there is no non-prime attribute that is transitively dependent on the primary key "Pharmacy_ID".

D. The **PHARMACY** relation schema satisfies all conditions of the BCNF because there exists no functional dependency X → A where X is not a super key or A is a prime attribute and X not a super key.

10. Allergy



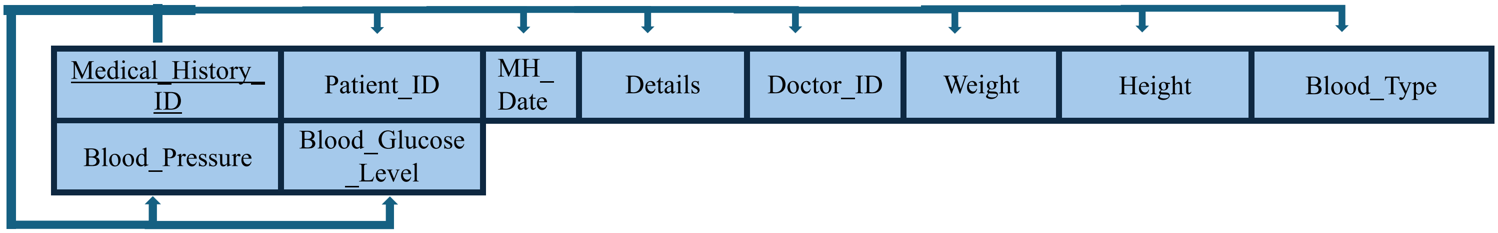
A. The **ALLERGY** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.

B. The **ALLERGY** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key "**Allergy_ID**".

C. The **ALLERGY** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there is no non-prime attribute that is transitively dependent on the primary key "**Allergy_ID**".

D. The **ALLERGY** relation schema satisfies all conditions of the BCNF because there exists no functional dependency X → A where X is not a super key or A is a prime attribute and X not a super key.

11. Medical_History



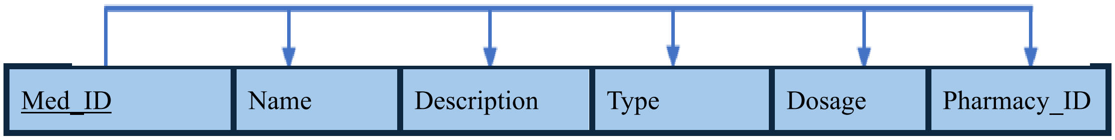
A. The **MEDICAL_HISTORY** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.

B. The **MEDICAL_HISTORY** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key “Medical_History_ID”.

C. The **MEDICAL_HISTORY** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there is no non-prime attribute that is transitively dependent on the primary key “Medical_History_ID”.

D. The **MEDICAL_HISTORY** relation schema satisfies all conditions of the BCNF because there exists no functional dependency X→A where X is not a super key or A is a prime attribute and X not a super key.

12. Medication



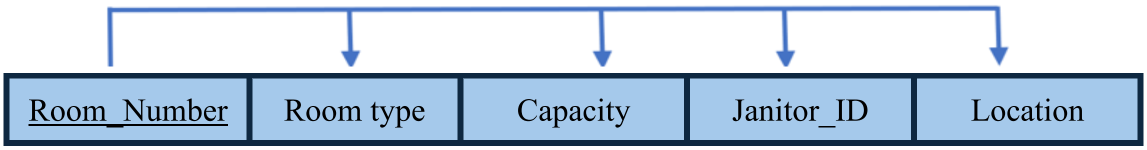
A. The **MEDICATION** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.

B. The **MEDICATION** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key “Med_ID”.

C. The **MEDICATION** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there is no non-prime attribute that is transitively dependent on the primary key “Med _ID”.

D. The **MEDICATION** relation schema satisfies all conditions of the BCNF because there exists no functional dependency X→A where X is not a super key or A is a prime attribute and X not a super key.

13. Room



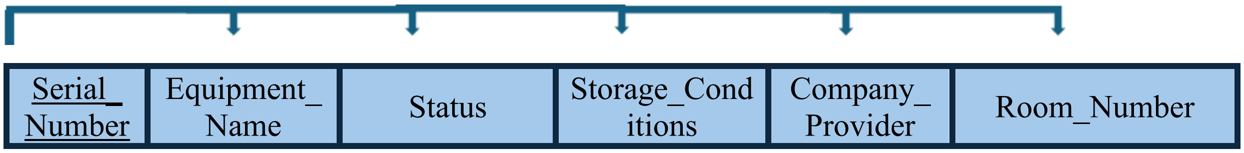
A. The **ROOM** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.

B. The **ROOM** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key "**Room_Number**".

C. The **ROOM** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there is no non-prime attribute that is transitively dependent on the primary key "**Room_Number**".

D. The **ROOM** relation schema satisfies all conditions of the BCNF because there exists no functional dependency X → A where X is not a super key or A is a prime attribute and X not a super key

14. Medical_Equipment



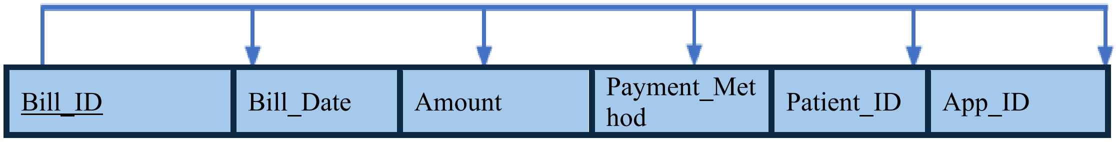
A. The **MEDICAL_EQUIPMENT** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.

B. The **MEDICAL_EQUIPMENT** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key "Serial_Number".

C. The **MEDICAL_EQUIPMENT** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there is no non-prime attribute that is transitively dependent on the primary key "Serial_Number".

D. The **MEDICAL_EQUIPMENT** relation schema satisfies all conditions of the BCNF because there exists no functional dependency X→A where X is not a super key or A is a prime attribute and X not a super key.

15. Billing



Bill_ID	Bill_Date	Amount	Payment_Method	Patient_ID	App_ID

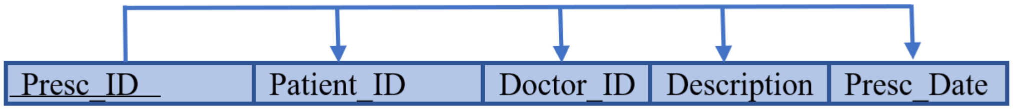
A. The **BILLING** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.

B. The **BILLING** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key "Bill_ID".

C. The **BILLING** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there is no non-prime attribute that is transitively dependent on the primary key "Bill_ID".

D. The **BILLING** relation schema satisfies all conditions of the BCNF because there exists no functional dependency X→A where X is not a super key or A is a prime attribute and X not a super key.

16. Prescription



Presc_ID	Patient_ID	Doctor_ID	Description	Presc_Date

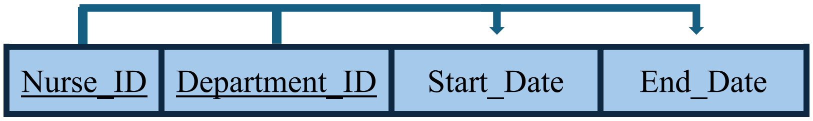
A. The **PRESCRIPTION** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.

B. The **PRESCRIPTION** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key **"Presc_ID, Patient_ID, Doctor_ID".**

C. The **PRESCRIPTION** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there is no non-prime attribute that is transitively dependent on the primary key **"Presc_ID, Patient_ID, Doctor_ID".**

D. The **PRESCRIPTION** relation schema satisfies all conditions of the BCNF because there exists no functional dependency X → A where X is not a super key or A is a prime attribute and X not a super key.

17. Belongs_To2



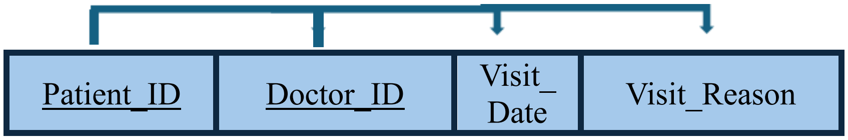
A. The **BELONGS_TO2** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.

B. The **BELONGS_TO2** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the primary key "Nurse_ID" and "Department_ID".

C. The **BELONGS_TO2** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there is no non-prime attributes that are transitively dependent on the primary key "Nurse_ID" and "Department_ID".

D. The **BELONGS_TO2** relation schema satisfies all conditions of the BCNF because there exists no functional dependency X→A where X is not a super key or A is a prime attribute and X not a super key.

18. Visits



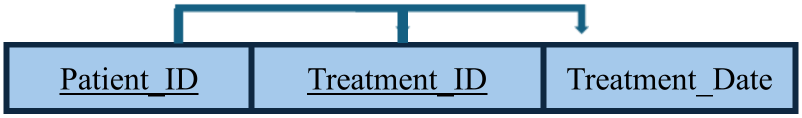
A. The **VISITS** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.

B. The **VISITS** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the composite primary key "Patient_ID" and "Doctor_ID".

C. The **VISITS** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there is no non-prime attribute that is transitively dependent on the primary key "Patient_ID" and "Doctor_ID".

D. The **VISITS** relation schema satisfies all conditions of the BCNF because there exists no functional dependency X→A where X is not a super key or A is a prime attribute and X not a super key.

19. Undergoes



A. The **UNDERGOES** relation schema satisfies all conditions of the 1NF because it has neither multivalued attributes nor composite attributes. All attributes are single and atomic.

B. The **UNDERGOES** relation schema satisfies all conditions of the 2NF because every nonprime attribute is fully functionally dependent on the composite primary key "Patient_ID", "Treatment_ID".

C. The **UNDERGOES** relation schema satisfies all conditions of the 3NF because it satisfies the 2NF and there is no non-prime attribute that is transitively dependent on the primary key "Patient_ID", "Treatment_ID".

D. The **UNDERGOES** relation schema satisfies all conditions of the BCNF because there exists no functional dependency X→A where X is not a super key or A is a prime attribute and X not a super key.

Relation Schemas without non-prime attributes:

20. Services

Nurse_ID	Patient_ID

21. Requires

Test_ID	Diagnosis_ID

22. Contains1

Presc_ID	Med_ID

23. Treats

Doctor_ID	Patient_ID

24. Doctor_Phone_Number

Doctor_ID	Phone_Number

25. Nurse_Phone_Number

Nurse_ID	Phone_Number

26. Patient_Phone_Number

Patient_ID	Phone_Number

XVI. Conclusion:

In this report, we have tried to accurately and efficiently model how a professional and high-quality clinic database must be designed, defined, and implemented. This project was divided into four phases. In Phase I, the entities and relationships that make up the database were designed. In Phase II, the higher-level descriptions were converted into relation schemas. In Phase III, the relation schemas were converted into SQL code and the tables and queries were built on the Oracle server. In the latest phase, Phase IV, the database was normalized.