BSc (Hons) Artificial Intelligence and Data Science

Module: CM1603 Database Systems

Individual Coursework Report

Module Leader: Ms. Dileeka Alwis

RGU Student ID : 2425472

IIT Student ID : 20240892

Student Name : Lidiya Rajapakse

Table of Contents

Section 1-Extended Entity Relationship Diagram (EERD)

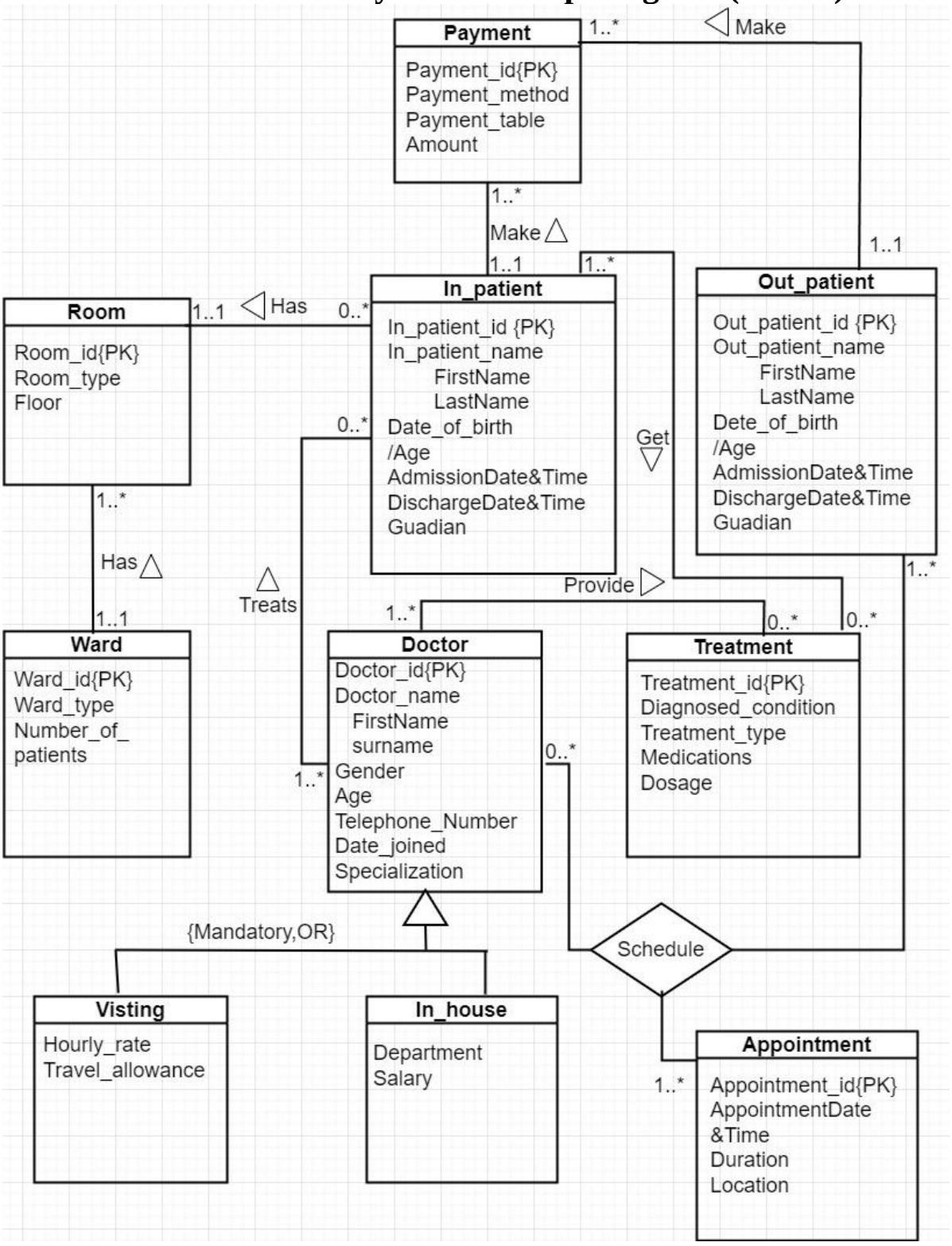


Figure 1:Conceptual Diagram

Assumptions:

- An out-patient should have at least 1 payment and can have many payments. One payment has only one out patient.
- An in-patient should have at least 1 payment and can have many payments. One payment has only one in patient.
- A patient has one room. A room may or may not have patients and multiple patients can share 1 room (Semi private rooms).
- Ward has at least 1 room and can have many rooms. One room belongs to only 1 ward.
- A doctor may or may not have patients and 1 doctor can treat multiple patients. An in-patient should at least have one doctor and can channel many doctors.
- An in-patient should have at least one treatment and can get multiple treatments. A treatment may or may not have patients and can have many patients.
- A doctor may or may not provide any treatments and a doctor can provide multiple treatments. A treatment should at least have 1 patient and can have multiple doctors.
- Doctors at GVH can be classified as visiting doctors and in-house doctors.
- Out-patients can schedule appointments with doctors to receive medical treatments.
- An outpatient should have an appointment and a doctor and can have many appointments and many doctors.
- A doctor may or may not have patients or appointments and a doctor can have many appointments and many outpatients.
- An appointment can only have one doctor and one outpatient.

Section 2-Logocal ERD (Relational Schema)

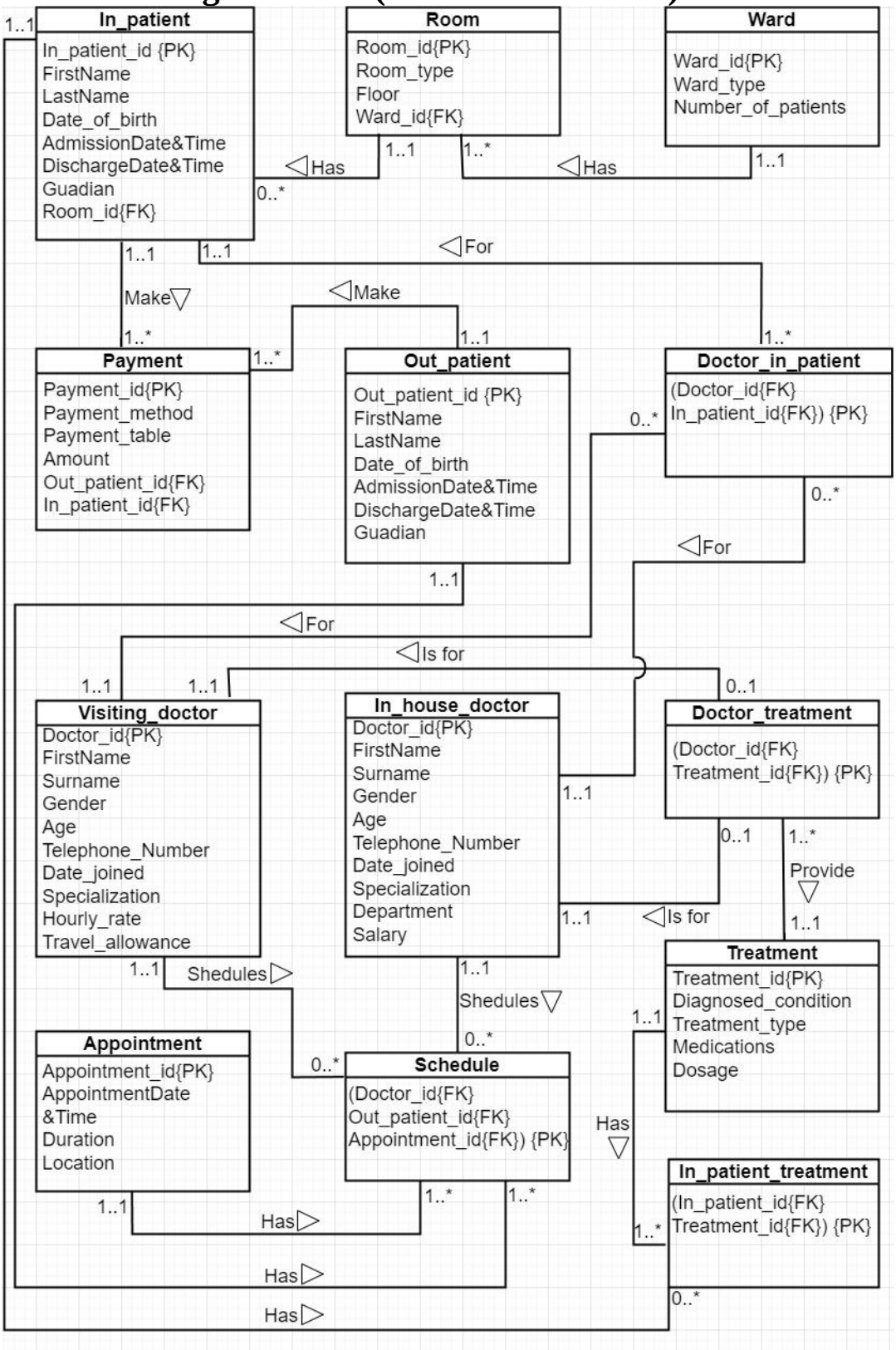


Figure 2:Logical Diagram

Assumptions:

- The hospital handles two types of patients: In-patients (admitted) and Out-patients.
- An in-patient is assigned to a room, which is linked to a ward. A ward can accommodate multiple rooms.
- Each room belongs to one ward, and a ward can contain multiple rooms.
- Doctors are categorized into Visiting doctors and In-house doctors and doctors may treat both in-patients and out-patients, tracked through Doctor_in_patient and Schedule.
- Each treatment is linked to a doctor, and one treatment can involve multiple visits or schedules.

Relational Schema

1. In_patient(In_patient_id{PK},FirstName,LastName,Date_of_birth,AdmissionDate&Time,DischargeDate&Time,Guadian,Room_id{FK})
2. Room (Room_id{PK},Room_type,Floor,Ward_id{FK})
3. Ward (Ward_id{PK},Ward_type,Number_of_patients)
4. Payment(Payment_id{PK},Payment_method,Amount,Out_patient_id{FK},In_patient_id{FK})
5. Out_patient(Out_patient_id{PK},FirstName,LastName,Date_of_birth,AdmissionDate&Time,DischargeDate&Time,Guadian)
6. Doctor_in_patient ((Doctor_id{FK},In_patient_id{FK}) {PK})
7. Visiting_doctor(Doctor_id{PK},FirstName,Surname,Gender,Age,Telephone_Number,Date_joined,Hourly_rate,Travel_allowance,Specialization)
8. In_house_doctor(Doctor_id{PK},FirstName,Surname,Gender,Age,Telephone_Number,Date_joined,Department,Salary,Specialization)
9. Doctor_treatment((Doctor_id{FK},Treatment_id{FK}) {PK})
10. Appointment (Appointment_id{PK},AppointmentDate&Time,Duration,Location)
11. Schedule ((Doctor_id{FK},Out_patient_id{FK},Appointment_id{FK}) {PK})
12. Treatment(Treatment_id{PK},Diagnosed_condition,Treatment_type,Medications,Dosage)
13. In_patient_treatment((In_patient_id{FK},Treatment_id{FK}){PK})

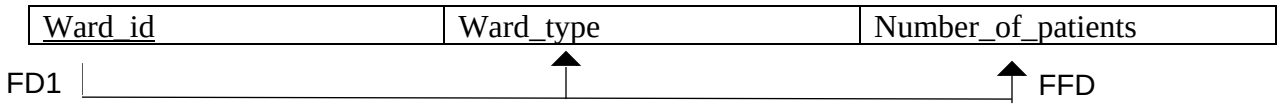
Section 3-Data Normalization (up to 3NF)

R1= (Ward_id,Ward_type,Number_of_patients)

Assumptions:

- PKs are underlined and FKs are dotted underlined.
- PK in R1 is Ward_id .

Dependencies:



FD1- Ward_id → (Ward_type,Number_of_patients)-FFD

1NF:

- Each column contains only one value for each row.
- There are no repeating groups ,composite attributes or derived attributes therefore R1 is already in 1NF.

R1= (Ward_id,Ward_type,Number_of_patients)

2NF:

- Every non-prime attribute is fully functionally dependent on the primary key.There are no partial dependencies.
- Ward_type and Number_of_patients are fully functionally dependent on Ward_id because neither of them depends on only a part of the key.
- Therefore R1 is already in 2NF.

R1= (Ward_id,Ward_type,Number_of_patients)

3NF:

- There are no transitive dependencies.
- Therefore R1 is already in 3NF.

R1= (Ward_id,Ward_type,Number_of_patients)

Ward= (Ward_id,Ward_type,Number_of_patients)

R2=(Out_patient_id,FirstName,LastName,Date_of_birth,Admission Date&Time,DischargeDate&Time,Guadian)

Assumptions:

- PKs are underlined and FKs are dotted underlined.
- PK in R2 is Out_patient_id.

Dependencies:

Out_patient_ Id	FirstName	LastName	Date_of_ birth	AdmissionDate &Time	Discharge Date&Time	Guadian

Dependencies:

FD1- Out_patient_id → (FirstName,LastName,Date_of_birth,AdmissionDate&Time, DischargeDate&Time,Guadian) -FFD

1NF:

- Each column contains only one value for each row.
- There are no repeating groups ,composite attributes or derived attributes therefore R2 is already in 1NF.

R2=(Out_patient_id,FirstName,LastName,Date_of_birth,AdmissionDate&Time,DischargeDate &Time,Guadian)

2NF:

- Every non-prime attribute is fully functionally dependent on the primary key.There are no partial dependencies.
- Therefore R2 is already in 2NF.

R2=(Out_patient_id,FirstName,LastName,Date_of_birth,AdmissionDate&Time,DischargeDate &Time,Guadian)

3NF:

- There are no transitive dependencies.
- Therefore R2 is already in 3NF.

R1=(Out_patient_id,FirstName,LastName,Date_of_birth,AdmissionDate&Time,DischargeDate &Time,Guadian)

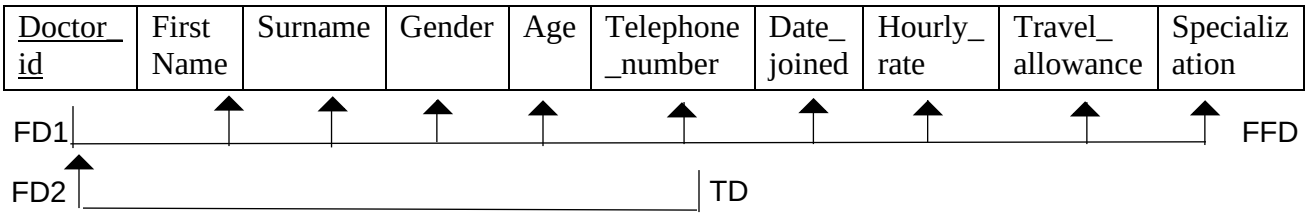
Out_patient=(Out_patient_id,FirstName,LastName,Date_of_birth,AdmissionDate&Time,Discha rgeDate&Time,Guadian)

R3=(Doctor_id,Doctor_name,Gender,Age,Telephone_Number,Date _joined,Hourly_rate,Travel_allowance,specialization)

Assumptions:

- PKs are underlined and FKs are dotted underlined.
- PK in R3 is Doctor_id.
- Telephone number is unique for a patient, and indirectly, other attributes depend on Telephone_Number.

Dependencies:



FD1-
Doctor_id→FirstName,Surname,Gender,Age,Telephone_Number,Date_joined,Hourly_rate, Travel_allowance ,specialization-FFD

FD2- Doctor_id→Telephone_Number→Doctor_id-TD

1NF:

- Each column contains only one value for each row.
- There are no repeating groups ,composite attributes or derived attributes therefore R3 is already in 1NF.

R3=(Doctor_id,FirstName,Surname,Gender,Age,Telephone_Number,Date_joined,Hourly_rate,T ravel_allowance,specialization)

2NF:

- Every non-prime attribute is fully functionally dependent on the primary key.There are

no partial dependencies.

- Therefore R3 is already in 2NF.

R3=(Doctor_id,FirstName,Surname,Gender,Age,Telephone_Number,Date_joined,Hourly_rate,Travel_allowance,specialization)

3NF:

- FD2 is a TD therefore R3 is not in 3NF.

R3=(Doctor_id,FirstName,Surname,Gender,Age,Date_joined,Hourly_rate,Travel_allowance,specialization)
R31=(Doctor_id,Telephone_number)

- Doctor_id and Telephone_number together form a primary key, allowing each doctor to have multiple telephone numbers.
- No transitive dependencies remain, Therefore R3 is in 3NF.

Visting_doctor=(Doctor_id,FirstName,Surname,Gender,Age,Date_joined,Hourly_rate,Travel_allowance,specialization)
Visting_doctor_telephone=(Doctor_id,Telephone_number)

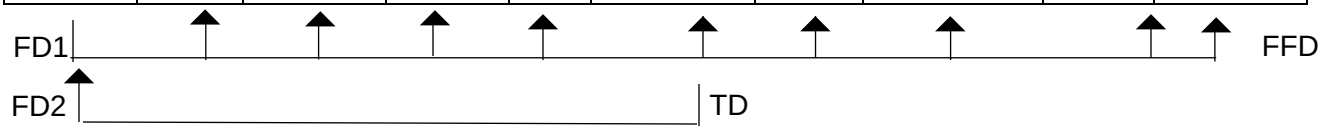
R4=(Doctor_id,FirstName,Surname,Gender,Age,Telephone_Number,Date_joined,Department,Salary,Specialization)

Assumptions:

- PKs are underlined and FKs are dotted underlined.
- PK in R4 is Doctor_id.
- Telephone number is unique for a patient, and indirectly, other attributes depend on Telephone_Number.

Dependencies:

Doctor_id	First Name	Surname	Gender	Age	Telephone_number	Date_joined	Department	Salary	Specialization



FD1-Doctor_id
→FirstName,Surname,Gender,Age,Telephone_Number,Date_joined,Department,Salary,specialization -FFD

FD2- Doctor_id→Telephone_Number→Doctor_id-TD

1NF:

- Each column contains only one value for each row.
- There are no repeating groups ,composite attributes or derived attributes therefore R4 is already in 1NF.

R4=(Doctor_id,FirstName,Surname,Gender,Age,Telephone_Number,Date_joined,Department,Salry,specialization)

2NF:

- Every non-prime attribute is fully functionally dependent on the primary key.There are no partial dependencies.
- Therefore R4 is already in 2NF.

R4=(Doctor_id,FirstName,Surname,Gender,Age,Telephone_Number,Date_joined,Department,Salary,specialization)

3NF:

- FD2 is a TD therefore R4 is not in 3NF.

R4=(Doctor_id,FirstName,Surname,Gender,Age,Telephone_Number,Date_joined,Department,Salary,specialization)
R41=(Doctor_id,Telephone_number)

- No transitive dependencies remain, Therefore R3 is in 3NF.

In_house_doctor=(Doctor_id,FirstName,Surname,Gender,Age,Date_joined,Department,Salary,specialization)
In_house_doctor_telephone=(Doctor_id,Telephone_number)

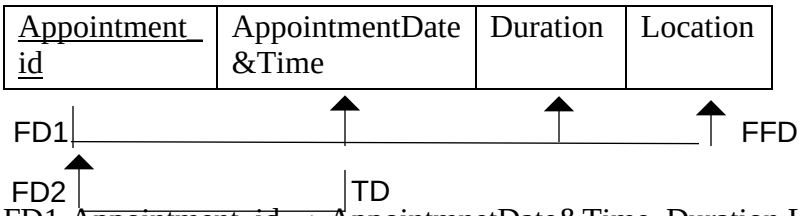
R5=(Appointment_id,AppointmentDate&Time,Duration,Location)

Assumptions:

- PKs are underlined and FKs are dotted underlined.

- PK in R5 is Appointment_id.

Dependencies:



FD1-Appointment_id → AppointmnetDate&Time ,Duration,Location-FFD

FD2- AppointmnetDate&Time→Appointmnet_id-TD

1NF:

- Each column contains only one value for each row.
- There are no repeating groups ,composite attributes or derived attributes therefore R5 is already in 1NF.

R5=(Appointment_id,AppointmentDate&Time,Duration,Location)

2NF:

- Every non-prime attribute is fully functionally dependent on the primary key.There are no partial dependencies.
- Therefore R5 is already in 2NF.

R5=(Appointment_id,AppointmentDate&Time,Duration,Location)

3NF:

- FD2 is a TD ,therefore R5 is not in 3NF.
- AppointmentDate&Time→Appointment_id, could be used to determine DurationDurationDuration and LocationLocationLocation.

R5=(Appointment_id,Duration,Location)
R51=(Appointment_id, AppointmentDate&Time)

- No transitive dependencies remain, Therefore R3 is in 3NF.

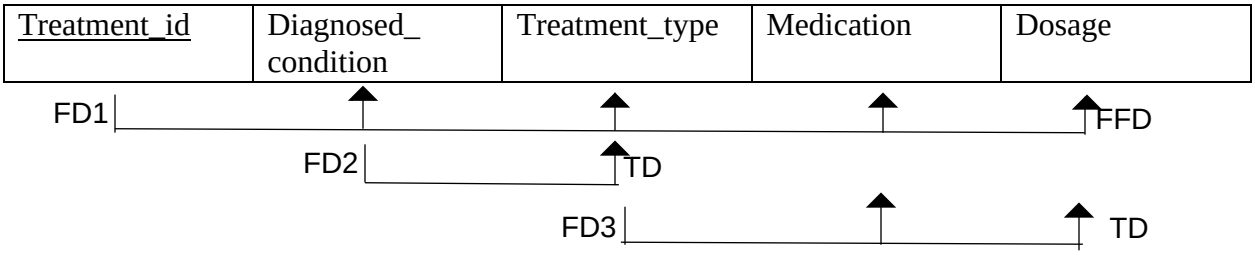
Appointmnet=(Appointment_id,Duration,Location)
Apointment_schedule=(Appointment_id, AppointmentDate&Time)

R6=(Treatment_id,Diagnosed_condition,Treatment_type,Medications,Dosage)

Assumptions:

- PKs are underlined and FKs are dotted underlined.
- PK in R6 is Treatment_id.

Dependencies:



FD1-Treatment_id→Diagnosed_condition,Treatment_type,Medications,Dosage-FFD

FD2-Diagnosed_condition→Treatment_type-TD

FD3-Treatment_type→Medications,Dosage-TD

1NF:

- Each column contains only one value for each row.
- There are no repeating groups ,composite attributes or derived attributes therefore R6 is already in 1NF.

R6=(Treatment_id,Diagnosed_condition,Treatment_type,Medications,Dosage)

2NF:

- Every non-prime attribute is fully functionally dependent on the primary key.There are no partial dependencies.
- Therefore R6 is already in 2NF.

R6=(Treatment_id,Diagnosed_condition,Treatment_type,Medications,Dosage)

3NF:

- FD2 and FD3 are TDs therefore R6 is not in 3NF.

R6(Treatment_id,Diagnosed_condition)
R61(Diagnosed_condition,Treatment_type)
R62(Treatment_type,Medications,Dosage)

- No transitive dependencies remain, Therefore R3 is in 3NF.

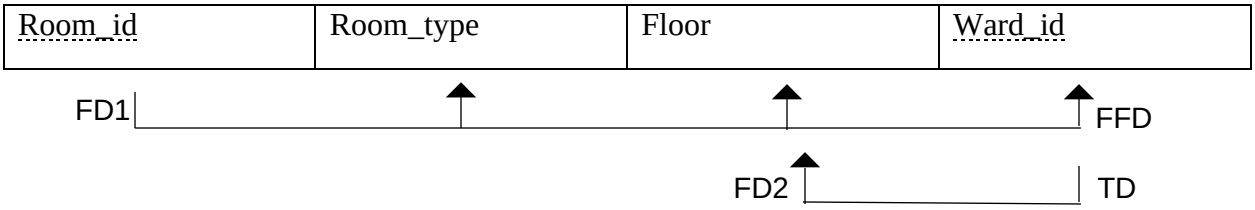
Treatment(Treatment_id,Diagnosed_condition)
Diagnosed_condition(Diagnosed_condition,Treatment_type)
Treatment_type(Treatment_type,Medications,Dosage)

R7= (Room_id,Room_type,Floor,Ward_id)

Assumptions:

- PKs are underlined and FKs are dotted underlined.
- PK in R7 is Room_id.
- FK in R7 is Ward_id.

Dependencies:



FD1-Room_id→Room_type,Floor,Ward_id-FFD

FD2-Ward_id→Floor-TD

1NF:

- Each column contains only one value for each row.
- There are no repeating groups ,composite attributes or derived attributes therefore R7 is already in 1NF.

R7= (Room_id,Room_type,Floor,Ward_id)

2NF:

- Every non-prime attribute is fully functionally dependent on the primary key.There are no partial dependencies.
- Therefore R7 is already in 2NF.

R7= (Room_id,Room_type,Floor,Ward_id)

3NF:

- FD2 is a TD therefore R7 is not in 3NF.

R7(Room_id,Room_type,Ward_id)
R71(Ward_id,Floor)

- No transitive dependencies remain, Therefore R7 is in 3NF.

Room(Room_id,Room_type,Ward_id)
Ward_allocation(Ward_id,Floor)

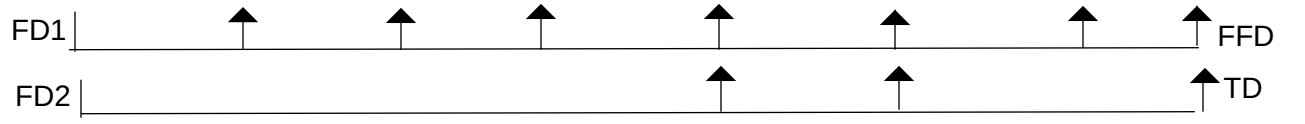
R8=(In_patient_id,FirstName,LastName,Date_of_birth,AdmissionDate&Time,DischargeDate&Time,Guadian,Room_id)

Assumptions:

- PKs are underlined and FKs are dotted underlined.
- PK in R8 is In_patient_id.
- FK in R8 is Room_id.

Dependencies:

In_patient_id	FirstName	LastName	Date_of_birth	Admission Date&Time	DischargeDate &Time	Guadian	Room_id



FD1-In_patient_id→FirstName,LastName,Date_of_birth,AdmissionDate&Time,
DischargeDate&Time,Guardian,Room_id-FFD

FD2-In_patient_id→AdmissionDate&Time,DischargeDate&Time,Room_id-TD

1NF:

- Each column contains only one value for each row.
- There are no repeating groups ,composite attributes or derived attributes therefore R8 is already in 1NF.

R8=(In_patient_id,FirstName,LastName,Date_of_birth,AdmissionDate&Time,DischargeDate&Time,Guadian,Room_id)

2NF:

- Every non-prime attribute is fully functionally dependent on the primary key.There are no partial dependencies.
- Therefore R8 is already in 2NF.

R8=(In_patient_id,FirstName,LastName,Date_of_birth,AdmissionDate&Time,DischargeDate&Time,Guadian,Room_id)

3NF:

- FD2 is a TD therefore R8 is not in 3NF.

R8(In_patient_id,FirstName,LastName,Date_of_birth,Guardian)
R81(In_patient_id,AdmissionDate&Time,DischargeDate&Time,Room_id)

- No transitive dependencies remain, Therefore R8 is in 3NF.

In_patient(In_patient_id,FirstName,LastName,Date_of_birth,Guardian)
In_patient_admission_details(In_patient_id,AdmissionDate&Time,DischargeDate&Time,Room_id)

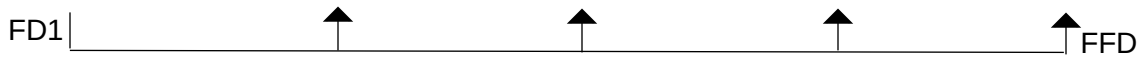
R9=(Payment_id,Payment_method,Amount,Out_patient_id,In_patient_id)

Assumptions:

- PKs are underlined and FKs are dotted underlined.
- PK in R9 is Payment_id.
- FKs in R9 are Out_patient_id and In_patient_id

Dependencies:

Payment_id	Payment_method	Amount	Out_patient_id	In_patient_id



FD1-Payment_id→Payment_method,Amount,Out_patient_id,In_patient_id-FFD

1NF:

- Each column contains only one value for each row.
- There are no repeating groups ,composite attributes or derived attributes therefore R9 is already in 1NF.

R9=(Payment_id,Payment_method,Amount,Out_patient_id,In_patient_id)

2NF:

- Every non-prime attribute is fully functionally dependent on the primary key.There are no partial dependencies.
- Therefore R9 is already in 2NF.

R9=(Payment_id,Payment_method,Amount,Out_patient_id,In_patient_id)

3NF:

- There are no transitive dependencies.
- Therefore R9 is already in 3NF.

R9=(Payment_id,Payment_method,Amount,Out_patient_id,In_patient_id)

Payment=(Payment_id,Payment_method,Amount,Out_patient_id,In_patient_id)

R10=((Doctor_id,In_patient_id))

Assumptions:

- PKs are underlined and FKs are dotted underlined.
- PK in R10 are Doctor_id and In_patient_id .
- FKs in R10 are Doctor_id and In_patient_id.

Dependencies:

Doctor_id	In_patient_id

FD1 ——————————————— FFD

FD1-(Doctor_id,In_patient_id)→(No other attributes)-FFD

1NF:

- Each column contains only one value for each row.
- There are no repeating groups ,composite attributes or derived attributes therefore R10 is already in 1NF.

R10=((Doctor_id,In_patient_id))

PK= (Doctor_id,In_patient_id)
FKs= Doctor_id
In_patient_id

2NF:

- Every non-prime attribute is fully functionally dependent on the primary key.There are no partial dependencies.
- Therefore R10 is already in 2NF.

R10=((Doctor_id,In_patient_id))

PK= (Doctor_id,In_patient_id)
FKs= Doctor_id
In_patient_id

3NF:

- There are no transitive dependencies.
- Therefore R10 is already in 3NF.

R10=((Doctor_id,In_patient_id))

PK= (Doctor_id,In_patient_id)
FKs= Doctor_id
In_patient_id

Doctor_in_patient=((Doctor_id,In_patient_id))

R11=((Doctor_id,Treatment_id))

Assumptions:

- PKs are underlined and FKs are dotted underlined.
- PK in R11 are Doctor_id and Treatment_id .
- FKs in R11 are Doctor_id and Treatment_id.

Dependencies:

Doctor_id	Treatment_id

FD1 ——————————————— FFD

FD1-(Doctor_id,Treatment_id)→(No other attributes)-FFD

1NF:

- Each column contains only one value for each row.
- There are no repeating groups ,composite attributes or derived attributes therefore R11 is already in 1NF.

R11=((Doctor_id,Treatment_id))

PK= (Doctor_id,Treatment_id)
FKs= Doctor_id
Treatment_id

2NF:

- Every non-prime attribute is fully functionally dependent on the primary key.There are no partial dependencies.
- Therefore R11 is already in 2NF.

R11=((Doctor_id,Treatment_id))

PK= (Doctor_id,Treatment_id)

FKs= Doctor_id
Treatment_id

3NF:

- There are no transitive dependencies.
- Therefore R11 is already in 3NF.

R11=((Doctor_id,Treatment_id))

PK= (Doctor_id,Treatment_id)
FKs= Doctor_id
Treatment_id

Doctor_treatment=((Doctor_id,Treatment_id))

R12=((In_patient_id,Treatment_id))

Assumptions:

- PKs are underlined and FKs are dotted underlined.
- PK in R12 are In-patient_id and Treatment_id .
- FKs in R12 are In_patient_id and Treatment_id.

Dependencies:

In_patient_id	Treatment_id

FD1 ——————————————— FFD

FD1-(In_patient_id,Treatment_id)→(No other attributes)-FFD

1NF:

- Each column contains only one value for each row.
- There are no repeating groups ,composite attributes or derived attributes therefore R12 is already in 1NF.

R12=((Doctor_id,Treatment_id))

PK= (In_patient_id,Treatment_id)

FKs= In_patient_id
Treatment_id

2NF:

- Every non-prime attribute is fully functionally dependent on the primary key.There are no partial dependencies.
- Therefore R12 is already in 2NF.

R12=((In_patient_id,Treatment_id))

PK= (In_patient_id,Treatment_id)
FKs= In_patient_id
Treatment_id

3NF:

- There are no transitive dependencies.
- Therefore R12 is already in 3NF.

R12=((In_patient_id,Treatment_id))

PK= (In_patient_id,Treatment_id)
FKs= In_patient_id
Treatment_id

In_patient_treatment=((In_patient_id,Treatment_id))

R13= ((Doctor_id,Out_patient_id,Appointment_id))

Assumptions:

- PKs are underlined and FKs are dotted underlined.
- PK in R13 are Doctor_id,Out_patient_id and Appointment_id .
- FKs in R12 are Doctor_id,Out_patient_id and Appointment_id.

Dependencies:

Doctor_id	Out_patient_id	Appointment_id

FD1 FFD

FD1-(Doctor_id, Out_patient_id ,Appointment_id)→(No other attributes)-FFD

1NF:

- Each column contains only one value for each row.
- There are no repeating groups ,composite attributes or derived attributes therefore R13 is already in 1NF.

R13=((Doctor_id, Out_patient_id ,Appointment_id))

PK= (Doctor_id, Out_patient_id ,Appointment_id)
FKs= Doctor_id
Out_patient_id
Appointment_id

2NF:

- Every non-prime attribute is fully functionally dependent on the primary key.There are no partial dependencies.
- Therefore R13 is already in 2NF.

R13=((Doctor_id, Out_patient_id ,Appointment_id))

PK= (Doctor_id, Out_patient_id ,Appointment_id)
FKs= Doctor_id
Out_patient_id
Appointment_id

3NF:

- There are no transitive dependencies.
- Therefore R13 is already in 3NF.

R13=((Doctor_id, Out_patient_id ,Appointment_id))

PK= (Doctor_id, Out_patient_id ,Appointment_id)
FKs= Doctor_id
Out_patient_id
Appointment_id

Schedule=((Doctor_id, Out_patient_id ,Appointment_id))

Section 4- Table Creation and Population of Data

1.Ward Table

Ward= (<u>Ward_id</u>,Ward_type,Number_of_patients)

#Creating ward table

```
CREATE TABLE Ward(
        Ward_id INT(10) NOT NULL,
        Ward_type Varchar(200) NOT NULL,
        Number_of_patients INT(200) NOT NULL,
        PRIMARY KEY(Ward_id)
);
```

#Inserting values to ward

```
INSERT INTO Ward
(Ward_id, Ward_type, Number_of_patients)
VALUES
(1, 'General', 15),
(2, 'ICU', 5),
(3, 'Pediatrics', 10),
(4, 'Surgery', 8),
(5, 'Maternity', 12);
```

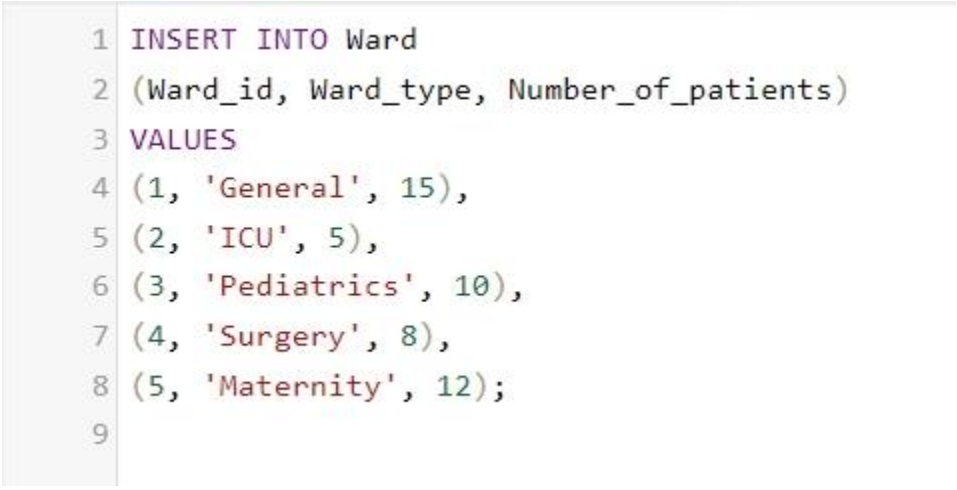


Figure 3:Insert into ward

```sql
CREATE TABLE Ward(
Ward_id INT(10) NOT NULL,
Ward_type Varchar(200) NOT NULL,
Number_of_patients INT(200) NOT NULL,
PRIMARY KEY(Ward_id)
);
```

Figure 4:create table ward

	#	Name	Type	Collation	Attributes	Null	Default
☐	1	**Ward_id** 🔑	int(10)			No	*None*
☐	2	**Ward_type**	varchar(200)	utf8mb4_general_ci		No	*None*
☐	3	**Number_of_patients**	int(200)			No	*None*

Figure 5:ward structure

←T→			Ward_id	Ward_type	Number_of_patients
☐	Edit	Copy Delete	1	General	15
☐	Edit	Copy Delete	2	ICU	5
☐	Edit	Copy Delete	3	Pediatrics	10
☐	Edit	Copy Delete	4	Surgery	8
☐	Edit	Copy Delete	5	Maternity	12

Figure 6:Ward table

2. Out_patient Table

Out_patient=(Out_patient_id,FirstName,LastName,Date_of_birth,Admission DateTime,DischargeDateTime,Guadian)

#Creating Out_patient table

```
CREATE TABLE Out_patient(
   Out_patient_id INT(10) NOT NULL,
   FirstName VARCHAR(200) NOT NULL,
   LastName VARCHAR(200) NOT NULL,
   Date_of_birth DateTime(50) ,
   AdmissionDateTime DateTime(50),
   DischargeDateTime DateTime(50),
   Guadian VARCHAR(200) NOT NULL,
   PRIMARY KEY(Out_patient_id)
);
```

#Inserting values to Out_patient

```
INSERT INTO Out_patient (Out_patient_id, FirstName, LastName, Date_of_birth, AdmissionDateTime,
DischargeDateTime, Guardian)
VALUES
(1, 'John', 'Doe', '1985-03-15 00:00:00', '2024-12-01 10:00:00', '2024-12-05 14:00:00', 'Jane Doe'),
(2, 'Emily', 'Smith', '1990-06-22 00:00:00', '2024-12-02 11:30:00', '2024-12-06 09:00:00', 'Mark Smith'),
(3, 'Michael', 'Brown', '1978-11-08 00:00:00', '2024-12-03 15:00:00', NULL, 'Sarah Brown'),
(4, 'Sarah', 'Johnson', '2002-09-30 00:00:00', '2024-12-04 08:00:00', '2024-12-07 16:00:00', 'David
Johnson'),
(5, 'David', 'Williams', '1995-01-12 00:00:00', '2024-12-05 10:00:00', NULL, 'Emma Williams');
```

```
1  CREATE TABLE Out_patient(
2      Out_patient_id INT(10) NOT NULL,
3      FirstName VARCHAR(200) NOT NULL,
4      LastName VARCHAR(200) NOT NULL,
5      Date_of_birth DateTime(6) ,
6      AdmissionDateTime DateTime(6),
7      DischargeDateTime DateTime(6),
8      Guadian VARCHAR(200) NOT NULL,
9      PRIMARY KEY(Out_patient_id)
10 );
```

Figure 7:Create table out_patient

```sql
INSERT INTO Out_patient (Out_patient_id, FirstName, LastName,
Date_of_birth, AdmissionDateTime, DischargeDateTime, Guardian)
VALUES
(1, 'John', 'Doe', '1985-03-15 00:00:00', '2024-12-01 10:00:00', '2024-
12-05 14:00:00', 'Jane Doe'),
(2, 'Emily', 'Smith', '1990-06-22 00:00:00', '2024-12-02 11:30:00',
'2024-12-06 09:00:00', 'Mark Smith'),
(3, 'Michael', 'Brown', '1978-11-08 00:00:00', '2024-12-03 15:00:00',
NULL, 'Sarah Brown'),
(4, 'Sarah', 'Johnson', '2002-09-30 00:00:00', '2024-12-04 08:00:00',
'2024-12-07 16:00:00', 'David Johnson'),
(5, 'David', 'Williams', '1995-01-12 00:00:00', '2024-12-05 10:00:00',
NULL, 'Emma Williams');
```

Figure 8:Insert table Out_patient

	#	Name	Type	Collation	Attributes	Null	Default
☐	1	**Out_patient_id** 🔑	int(10)			No	*None*
☐	2	**FirstName**	varchar(200)	utf8mb4_general_ci		No	*None*
☐	3	**LastName**	varchar(200)	utf8mb4_general_ci		No	*None*
☐	4	**Date_of_birth**	datetime(6)			Yes	*NULL*
☐	5	**AdmissionDateTime**	datetime(6)			Yes	*NULL*
☐	6	**DischargeDateTime**	datetime(6)			Yes	*NULL*
☐	7	**Guardian**	varchar(200)	utf8mb4_general_ci		No	*None*

Figure 9:out_patient structure

←T→	Out_patient_id	FirstName	LastName	Date_of_birth	AdmissionDateTime	DischargeDateTime	Guardian
☐ Edit Copy Delete	1	John	Doe	1985-03-15 00:00:00.000000	2024-12-01 10:00:00.000000	2024-12-05 14:00:00.000000	Jane Doe
☐ Edit Copy Delete	2	Emily	Smith	1990-06-22 00:00:00.000000	2024-12-02 11:30:00.000000	2024-12-06 09:00:00.000000	Mark Smith
☐ Edit Copy Delete	3	Michael	Brown	1978-11-08 00:00:00.000000	2024-12-03 15:00:00.000000	*NULL*	Sarah Brown
☐ Edit Copy Delete	4	Sarah	Johnson	2002-09-30 00:00:00.000000	2024-12-04 08:00:00.000000	2024-12-07 16:00:00.000000	David Johnson
☐ Edit Copy Delete	5	David	Williams	1995-01-12 00:00:00.000000	2024-12-05 10:00:00.000000	*NULL*	Emma Williams

↑ ☐ Check all *With selected:* Edit Copy Delete Export

Figure 10:Out_patient table

3. Visting_doctor Table
Visting_doctor=(Doctor_id,Doctor_name,Gender,Age,Date_joined,Hourly_rate,Travel_allowance,specialization)

#Creating Visiting_doctor table

```
CREATE TABLE Visiting_doctor (
   Doctor_id INT(10) NOT NULL,
   FurstName VARCHAR(200),
   Surname VARCHAR(200),
   Gender VARCHAR(10) NOT NULL,
   Age INT(3) NOT NULL,
   Date_joined DATE NOT NULL,
   Hourly_rate DECIMAL(10, 2) NOT NULL,
   Travel_allowance DECIMAL(10, 2),
   Specialization VARCHAR(200) NOT NULL,
   PRIMARY KEY (Doctor_id)
);
```

#Inserting values to Visiting_doctor

```
INSERT INTO Visiting_doctor (Doctor_id, FirstName,Surname, Gender, Age, Date_joined, Hourly_rate,
Travel_allowance)
VALUES
(1, 'John',' Smith', 'Male', 45, '2015-06-12', 150.00, 20.00,Cardiology),
(2, 'Emily', 'Johnson', 'Female', 38, '2018-04-25', 175.00, 25.00,neurology),
(3, 'Alex', 'Taylor', 'Male', 50, '2010-09-15', 200.00, 30.00,Cardiology),
(4, 'Sarah', 'Peterson', 'Female', 29, '2020-01-20', 120.00, 15.00, Pediatric),
(5, 'Michael', 'Brown', 'Other', 34, '2019-11-10', 180.00, 0.00, Orthopaedics);
```

```
1  CREATE TABLE Visiting_doctor (
2      Doctor_id INT(10) NOT NULL,
3      FurstName VARCHAR(200),
4      Surname VARCHAR(200),
5      Gender VARCHAR(10) NOT NULL,
6      Age INT(3) NOT NULL,
7      Date_joined DATE NOT NULL,
8      Hourly_rate DECIMAL(10, 2) NOT NULL,
9      Travel_allowance DECIMAL(10, 2),
10     Specialization VARCHAR(200) NOT NULL,
11     PRIMARY KEY (Doctor_id)
12 );
```

Figure 11:Create Visiting_doctor

```
INSERT INTO Visiting_doctor (Doctor_id, FirstName,Surname, Gender, Age, Date_joined, Hourly_rate, Travel_allowance)
VALUES
(1, 'John',' Smith', 'Male', 45, '2015-06-12', 150.00, 20.00,Cardiology),
(2, 'Emily', 'Johnson', 'Female', 38, '2018-04-25', 175.00, 25.00,neurology),
(3, 'Alex', 'Taylor', 'Male', 50, '2010-09-15', 200.00, 30.00,Cardiology),
(4, 'Sarah', 'Peterson', 'Female', 29, '2020-01-20', 120.00, 15.00, Pediatric),
(5, 'Michael', 'Brown', 'Other', 34, '2019-11-10', 180.00, 0.00, Orthopaedics);

```

Figure 12:Insert into Visiting_doctor

	#	Name	Type	Collation	Attributes	Null	Default
☐	1	Doctor_id 🔑	int(10)			No	None
☐	2	FirstName	varchar(200)	utf8mb4_general_ci		No	None
☐	3	Surname	varchar(200)	utf8mb4_general_ci		No	None
☐	4	Gender	varchar(10)	utf8mb4_general_ci		No	None
☐	5	Age	int(3)			No	None
☐	6	Date_joined	date			No	None
☐	7	Hourly_rate	decimal(10,2)			No	None
☐	8	Travel_allowance	decimal(10,2)			Yes	NULL
☐	9	Specialization	varchar(200)	utf8mb4_general_ci		No	None

Figure 13:Visiting_doctor structure

←T→	Doctor_id	FirstName	Surname	Gender	Age	Date_joined	Hourly_rate	Travel_allowance	Specialization
☐ Edit Copy Delete	1	John	Smith	Male	45	2015-06-12	150.00	20.00	Cardiology
☐ Edit Copy Delete	2	Emily	Johnson	Female	38	2018-04-25	175.00	25.00	neurology
☐ Edit Copy Delete	3	Alex	Taylor	Male	50	2010-09-15	200.00	30.00	Cardiology
☐ Edit Copy Delete	4	Sarah	Peterson	Female	29	2020-01-20	120.00	15.00	Pediatric
☐ Edit Copy Delete	5	Michael	Brown	Other	34	2019-11-10	180.00	0.00	Orthopaedics

Figure 14:Visiting_Doctor table

4. Visting_doctor_telephone Table
Visting_doctor_telephone=(Doctor_id,Telephone_number)

#Creating Visiting_doctor_telephone table

```
CREATE TABLE Visiting_doctor_telephone (
        Doctor_id INT(10) NOT NULL,
        Telephone_number VARCHAR(15) NOT NULL,
        PRIMARY KEY (Doctor_id, Telephone_number),
        FOREIGN KEY (Doctor_id) REFERENCES Visiting_doctor(Doctor_id)
);
```

#Inserting values to Visiting_doctor_telephone

```
INSERT INTO Visiting_doctor_telephone (Doctor_id, Telephone_number)
VALUES
(1, '0723456789'), (1, '0778765432'), (2, '0112233445'), (3, '0778877665'), (4, '0775667789')
;
```

```
1 CREATE TABLE Visiting_doctor_telephone (
2 Doctor_id INT(10) NOT NULL,
3 Telephone_number VARCHAR(15) NOT NULL,
4 PRIMARY KEY (Doctor_id, Telephone_number),
5 FOREIGN KEY (Doctor_id) REFERENCES Visiting_doctor(Doctor_id)
6 );
```

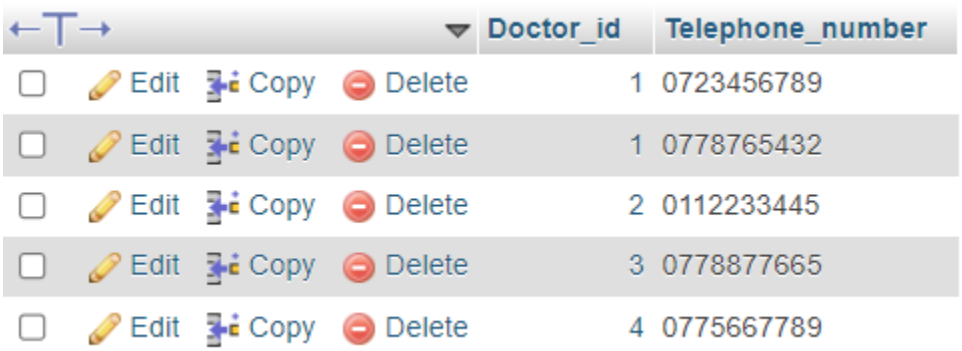
Figure 15:Create Visiting_doctor_telephone

```
1 #Inserting values to Visiting_doctor_telephone
2 INSERT INTO Visiting_doctor_telephone (Doctor_id, Telephone_number)
3 VALUES
4 (1, '0723456789'), (1, '0778765432'), (2, '0112233445'), (3, '0778877665'), (4, '0775667789')
5
```

Figure 16:Insert into Visiting_doctor_telephone

	#	Name	Type	Collation	Attributes	Null	Default
☐	1	**Doctor_id** 🔑	int(10)			No	*None*
☐	2	**Telephone_number** 🔑	varchar(15)	utf8mb4_general_ci		No	*None*

Figure 17:Visiting_doctor_telephone structure



	Doctor_id	Telephone_number
Edit Copy Delete	1	0723456789
Edit Copy Delete	1	0778765432
Edit Copy Delete	2	0112233445
Edit Copy Delete	3	0778877665
Edit Copy Delete	4	0775667789

Figure 18:Visiting_doctor_telephone

5. In_house_doctor Table

In_house_doctor=(Doctor_id,Doctor_name,Gender,Age,Date_joined,Department,Salary)

#Creating In_house_doctor table

```
CREATE TABLE In_house_doctor (
   Doctor_id INT(10) NOT NULL,
   FirstName VARCHAR(200),
   Surname VARCHAR(200),
   Gender VARCHAR(10) NOT NULL,
   Age INT(3) NOT NULL,
   Date_joined DATE NOT NULL,
   Department VARCHAR(100) NOT NULL,
   Salary DECIMAL(10, 2) NOT NULL,
   Specialization VARCHAR(200),
   PRIMARY KEY (Doctor_id)
);
```

#Inserting values to In_house_doctor

```
INSERT INTO In_house_doctor (Doctor_id, FirstName,surname, Gender, Age, Date_joined,
Department, Salary)
VALUES
(1, 'Alice', 'Brown', 'Female', 40, '2016-05-10', 'Cardiology', 75000.00,Cardiology),
(2, 'Robert', 'Smith', 'Male', 35, '2018-07-15', 'Neurology', 85000.00,Neurology),
(3, 'Karen', 'Johnson', 'Female', 45, '2012-10-22', 'Pediatrics', 78000.00,Cardiology),
(4, 'Peter', 'Lee', 'Male', 50, '2010-01-05', 'Orthopedics', 92000.00, Dermatology),
(5, 'Elton','Fedrick', 'Other', 32, '2020-03-18', 'Radiology', 68000.00, Pediatric);
```

```
1  CREATE TABLE In_house_doctor (
2      Doctor_id INT(10) NOT NULL,
3      FirstName VARCHAR(200),
4      Surname VARCHAR(200),
5      Gender VARCHAR(10) NOT NULL,
6      Age INT(3) NOT NULL,
7      Date_joined DATE NOT NULL,
8      Department VARCHAR(100) NOT NULL,
9      Salary DECIMAL(10, 2) NOT NULL,
10     Specialization VARCHAR(200),
11     PRIMARY KEY (Doctor_id)
12 );
```

Figure 19:Create In_house_doctor

```
INSERT INTO In_house_doctor (Doctor_id, FirstName,surname, Gender, Age, Date_joined, Department, Salary)
VALUES
(1, 'Alice', 'Brown', 'Female', 40, '2016-05-10', 'Cardiology', 75000.00,Cardiology),
(2, 'Robert', 'Smith', 'Male', 35, '2018-07-15', 'Neurology', 85000.00,Neurology),
(3, 'Karen', 'Johnson', 'Female', 45, '2012-10-22', 'Pediatrics', 78000.00,Cardiology),
(4, 'Peter', 'Lee', 'Male', 50, '2010-01-05', 'Orthopedics', 92000.00, Dermatology),
(5, 'Elton','Fedrick', 'Other', 32, '2020-03-18', 'Radiology', 68000.00, Pediatric);

```

Figure 20:Insert into In_house_doctor

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**Doctor_id**	int(10)			No	*None*			Change Drop More
☐	2	**FirstName**	varchar(200)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	3	**Surname**	varchar(200)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	4	**Gender**	varchar(10)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	5	**Age**	int(3)			No	*None*			Change Drop More
☐	6	**Date_joined**	date			No	*None*			Change Drop More
☐	7	**Department**	varchar(100)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	8	**Salary**	decimal(10,2)			No	*None*			Change Drop More
☐	9	**Specialization**	varchar(200)	utf8mb4_general_ci		No	*None*			Change Drop More

Figure 21:In house_doctor structure

←T→	Doctor_id	FirstName	Surname	Gender	Age	Date_joined	Department	Salary	Specialization
☐ Edit Copy Delete	1	Alice	Brown	Female	40	2016-05-10	Cardiology	75000.00	Cardiology
☐ Edit Copy Delete	2	Robert	Smith	Male	35	2018-07-15	Neurology	85000.00	Neurology
☐ Edit Copy Delete	3	Karen	Johnson	Female	45	2012-10-22	Pediatrics	78000.00	Cardiology
☐ Edit Copy Delete	4	Peter	Lee	Male	50	2010-01-05	Orthopedics	92000.00	Dermatology
☐ Edit Copy Delete	5	Elton	Fedrick	Other	32	2020-03-18	Radiology	68000.00	Pediatric

Figure 22:In house_doctor table

6. In_house_doctor_telephone Table
In_house_doctor_telephone=(Doctor_id,Telephone_number)

#Creating In_house_doctor table

```
CREATE TABLE In_house_doctor_telephone (
        Doctor_id INT(11) NOT NULL,
        Telephone_number VARCHAR(15) NOT NULL,
        PRIMARY KEY (Doctor_id, Telephone_number),
        FOREIGN KEY (Doctor_id) REFERENCES In_house_doctor(Doctor_id)
);
```

#Inserting values to In_house_doctor

```
INSERT INTO In_house_doctor_telephone (Doctor_id, Telephone_number)
VALUES
(1, '0774502162'), (1, '0714526893'), (2, '0754128963'), (3, '0774856791'), (4, '0721457859')
;
```

```
1 CREATE TABLE In_house_doctor_telephone (
2 Doctor_id INT NOT NULL,
3 Telephone_number VARCHAR(15) NOT NULL,
4 PRIMARY KEY (Doctor_id, Telephone_number),
5 FOREIGN KEY (Doctor_id) REFERENCES In_house_Doctor(Doctor_id)
6 );
7
```

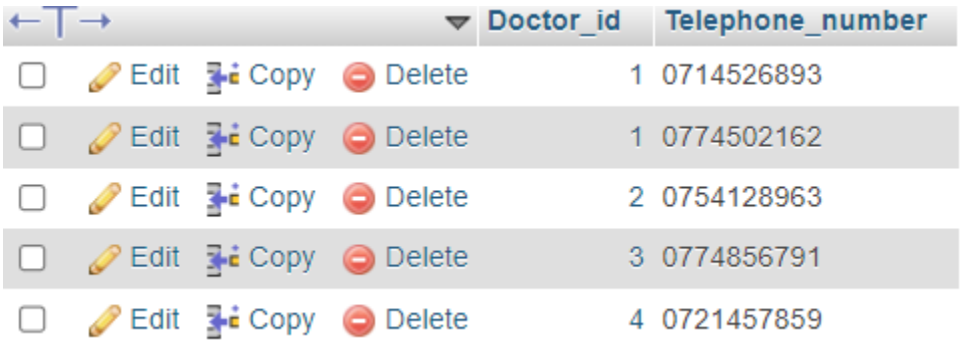
Figure 23:Create In_house_doctor_telephone

```
1 INSERT INTO In_house_doctor_telephone (Doctor_id, Telephone_number)
2 VALUES
3 (1, '0774502162'), (1, '0714526893'), (2, '0754128963'), (3, '0774856791'), (4, '0721457859')
4 ;
5
```

Figure 24:Insert into In_house_doctor_telephone

	#	Name	Type	Collation	Attributes	Null	Default
☐	1	**Doctor_id** 🔑	int(11)			No	*None*
☐	2	**Telephone_number** 🔑	varchar(15)	utf8mb4_general_ci		No	*None*

Figure 25:In_house_doctor_telephone structure



	Doctor_id	Telephone_number
Edit Copy Delete	1	0714526893
Edit Copy Delete	1	0774502162
Edit Copy Delete	2	0754128963
Edit Copy Delete	3	0774856791
Edit Copy Delete	4	0721457859

Figure 26:In_house_doctor_telephone

7. Appointmnet Table
Appointmnet=(Appointment_id,Duration,Location)

#Creating Appointment table

```
CREATE TABLE Appointment (
        Appointment_id INT(11) NOT NULL,
        Duration INT(11) NOT NULL,
        Location VARCHAR(100) NOT NULL,
        PRIMARY KEY(Appointment_id)
);
```

#Inserting values to Appointment

```
INSERT INTO Appointment (Appointment_id, Duration, Location)
VALUES
(101, 30, 'Room 1A'), (102, 45, 'Room 2B'), (103, 60, 'Room 3C'), (104, 90, 'Room 4D'), (105, 120,
'Room 5E')
;
```

```
1  CREATE TABLE Appointment (
2      Appointment_id INT NOT NULL,
3      Duration INT NOT NULL,
4      Location VARCHAR(100) NOT NULL,
5      PRIMARY KEY(Appointment_id)
6  );
7
```

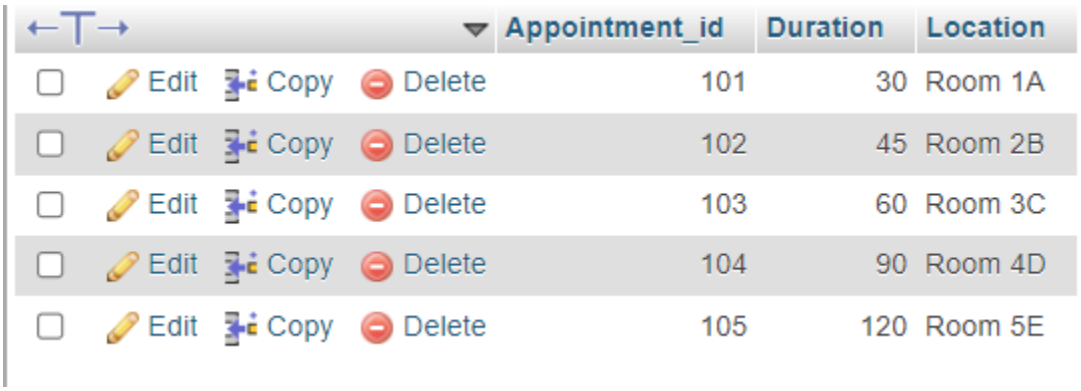
Figure 27:Create appointment

```
1  INSERT INTO Appointment (Appointment_id, Duration, Location)
2  VALUES
3  (101, 30, 'Room 1A'), (102, 45, 'Room 2B'), (103, 60, 'Room 3C'), (104, 90, 'Room 4D'), (105, 120,
   'Room 5E')
4  ;
5
```

Figure 28:Insert into Appointment

	#	Name	Type	Collation	Attributes	Null	Default
☐	1	**Appointment_id** 🔑	int(11)			No	*None*
☐	2	**Duration**	int(11)			No	*None*
☐	3	**Location**	varchar(100)	utf8mb4_general_ci		No	*None*

Figure 29:Appointment structure



	Appointment_id	Duration	Location
Edit Copy Delete	101	30	Room 1A
Edit Copy Delete	102	45	Room 2B
Edit Copy Delete	103	60	Room 3C
Edit Copy Delete	104	90	Room 4D
Edit Copy Delete	105	120	Room 5E

Figure 30:Appointment table

8. Apointment_schedule

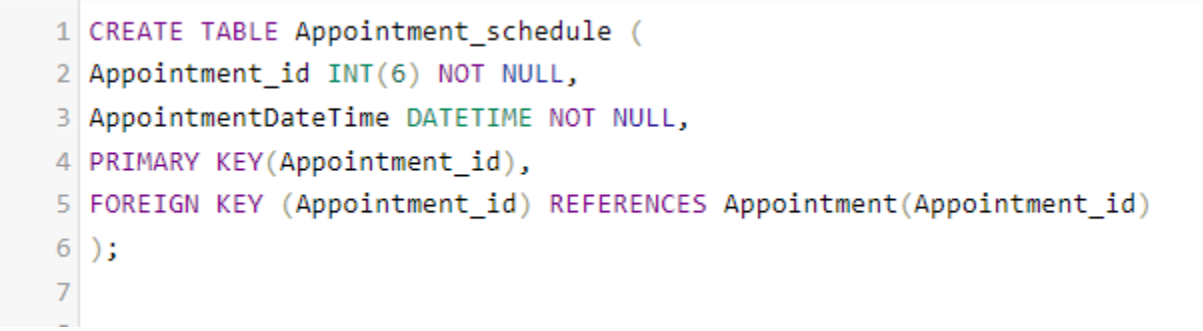
Apointment_schedule=(Appointment_id, AppointmentDate&Time)

#Creating Appointment_schedule table

```
CREATE TABLE Appointment_schedule (
        Appointment_id INT(6) NOT NULL,
        AppointmentDateTime DATETIME NOT NULL,
        PRIMARY KEY(Appointment_id),
        FOREIGN KEY (Appointment_id) REFERENCES Appointment(Appointment_id)
);
```

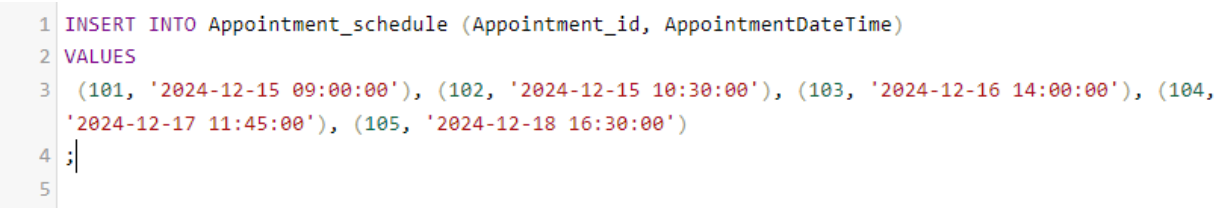
#Inserting values to Appointment_schedule

```
INSERT INTO Appointment_schedule (Appointment_id, AppointmentDateTime)
VALUES
 (101, '2024-12-15 09:00:00'), (102, '2024-12-15 10:30:00'), (103, '2024-12-16 14:00:00'), (104, '2024-12-17 11:45:00'), (105, '2024-12-18 16:30:00')
;
```



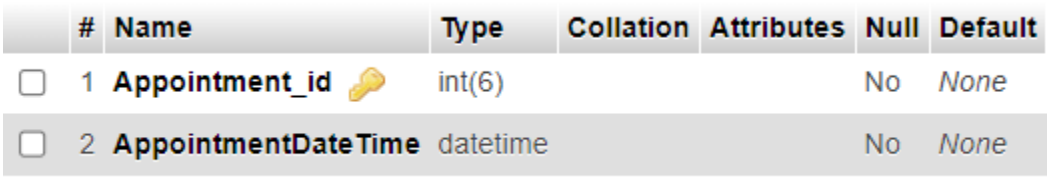
```
1 CREATE TABLE Appointment_schedule (
2 Appointment_id INT(6) NOT NULL,
3 AppointmentDateTime DATETIME NOT NULL,
4 PRIMARY KEY(Appointment_id),
5 FOREIGN KEY (Appointment_id) REFERENCES Appointment(Appointment_id)
6 );
7
```

Figure 31:Create Appointment_schedule



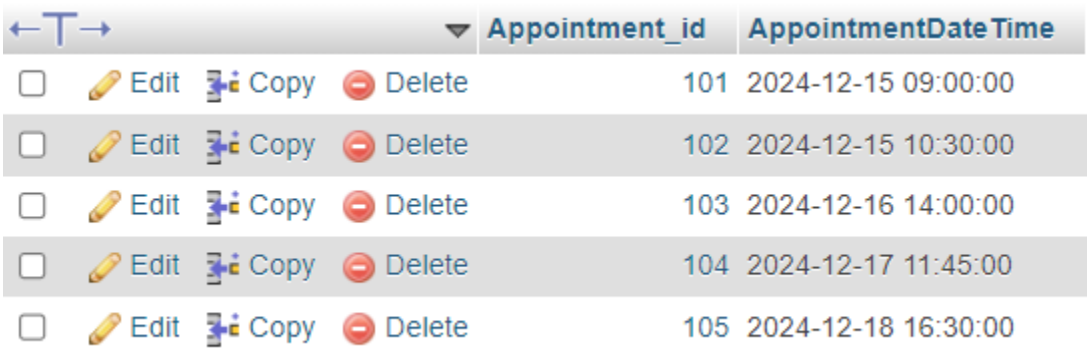
```
1 INSERT INTO Appointment_schedule (Appointment_id, AppointmentDateTime)
2 VALUES
3  (101, '2024-12-15 09:00:00'), (102, '2024-12-15 10:30:00'), (103, '2024-12-16 14:00:00'), (104,
  '2024-12-17 11:45:00'), (105, '2024-12-18 16:30:00')
4 ;
5
```

Figure 32:Insert into Appointment_schedule



	#	Name	Type	Collation	Attributes	Null	Default
☐	1	**Appointment_id** 🔑	int(6)			No	*None*
☐	2	**AppointmentDateTime**	datetime			No	*None*

Figure 33:Appointment_schedule structure



	Appointment_id	AppointmentDateTime
Edit Copy Delete	101	2024-12-15 09:00:00
Edit Copy Delete	102	2024-12-15 10:30:00
Edit Copy Delete	103	2024-12-16 14:00:00
Edit Copy Delete	104	2024-12-17 11:45:00
Edit Copy Delete	105	2024-12-18 16:30:00

Figure 34:Appointment_schedule table

9. Treatment table
Treatment(Treatment_id,Diagnosed_condition)

#Creating Treatment table
```
CREATE TABLE Treatment (
   Treatment_id INT(6) NOT NULL ,
   Diagnosed_condition VARCHAR(255) NOT NULL,
   PRIMARY KEY (Treatment_id )
);
```

#Inserting values to Treatment
```
INSERT INTO Treatment (Treatment_id, Diagnosed_condition)
VALUES
(1, 'Hypertension'), (2, 'Diabetes Type 2'), (3, 'Asthma'), (4, 'Migraine'), (5, 'Arthritis')
;
```

```
1 CREATE TABLE Treatment (
2     Treatment_id INT(6) NOT NULL ,
3     Diagnosed_condition VARCHAR(255) NOT NULL,
4     PRIMARY KEY (Treatment_id )
5 );
6
```

Figure 35: Create treatment

```
1 INSERT INTO Treatment (Treatment_id, Diagnosed_condition)
2 VALUES
3 (1, 'Hypertension'), (2, 'Diabetes Type 2'), (3, 'Asthma'), (4, 'Migraine'), (5, 'Arthritis')
4 ;
5
6
```

Figure 36:Insert into treatment

	#	Name	Type	Collation	Attributes	Null	Default
☐	1	**Treatment_id** 🔑	int(6)			No	*None*
☐	2	**Diagnosed_condition**	varchar(255)	utf8mb4_general_ci		No	*None*

Figure 37:treatment structure

	Treatment_id	Diagnosed_condition
Edit Copy Delete	1	Hypertension
Edit Copy Delete	2	Diabetes Type 2
Edit Copy Delete	3	Asthma
Edit Copy Delete	4	Migraine
Edit Copy Delete	5	Arthritis

Figure 38:treatment table

10. Diagnosed_condition table
Diagnosed_condition(Diagnosed_condition,Treatment_type)

#Creating Diagnosed_condition table
```
CREATE TABLE Diagnosed_condition (
   Diagnosed_condition VARCHAR(255) NOT NULL,
   Treatment_type VARCHAR(100) NOT NULL,
   PRIMARY KEY(Diagnosed_condition)
);
```

#Inserting values to Diagnosed_condition
```
INSERT INTO Diagnosed_condition (Diagnosed_condition, Treatment_type)
VALUES
('Hypertension', 'Medication'),
('Diabetes Type 2', 'Lifestyle Management and Medication'),
('Asthma', 'Inhalers and Medication'),
('Migraine', 'Pain Management'),
('Arthritis', 'Physical Therapy and Medication')
;
```

```
1  CREATE TABLE Diagnosed_condition (
2      Diagnosed_condition VARCHAR(255) NOT NULL,
3      Treatment_type VARCHAR(100) NOT NULL,
4      PRIMARY KEY(Diagnosed_condition)
5  );
6
```

Figure 39:Create diagnosed condition

```
1  INSERT INTO Diagnosed_condition (Diagnosed_condition, Treatment_type)
2  VALUES
3  ('Hypertension', 'Medication'),
4  ('Diabetes Type 2', 'Lifestyle Management and Medication'),
5  ('Asthma', 'Inhalers and Medication'),
6  ('Migraine', 'Pain Management'),
7  ('Arthritis', 'Physical Therapy and Medication')
8  ;
9
```

Figure 40:Insert into doagnosed condtion

	#	Name	Type	Collation	Attributes	Null	Default
☐	1	**Diagnosed_condition**	varchar(255)	utf8mb4_general_ci		No	*None*
☐	2	**Treatment_type**	varchar(100)	utf8mb4_general_ci		No	*None*

Figure 41:diagnosed condition structure

	Diagnosed_condition	Treatment_type
☐ Edit Copy Delete	Arthritis	Physical Therapy and Medication
☐ Edit Copy Delete	Asthma	Inhalers and Medication
☐ Edit Copy Delete	Diabetes Type 2	Lifestyle Management and Medication
☐ Edit Copy Delete	Hypertension	Medication
☐ Edit Copy Delete	Migraine	Pain Management

Figure 42:diagnosed condition table

11. Treatment_type table
Treatment_type(Treatment_type,Medications,Dosage)

#Creating Treatment_type table

```
CREATE TABLE Treatment_type (
    Treatment_type VARCHAR(100) NOT NULL,
    Medications VARCHAR(255) NOT NULL,
    Dosage VARCHAR(100) NOT NULL,
    PRIMARY KEY(Treatment_type)
);
```

#Inserting values to Treatment_type

```
INSERT INTO Treatment_type (Treatment_type, Medications, Dosage)
VALUES
('Medication', 'Amlodipine', '5mg once daily'),
('Lifestyle Management and Medication', 'Metformin', '500mg twice daily'),
('Surgery', 'Salbutamol', '2 puffs as needed before surgery'),
('Pain Management', 'Ibuprofen', '200mg every 6 hours as needed'),
('Physical Therapy and Medication', 'Paracetamol', '500mg every 4-6 hours')
;
```

```
1  CREATE TABLE Treatment_type (
2      Treatment_type VARCHAR(100) NOT NULL,
3      Medications VARCHAR(255) NOT NULL,
4      Dosage VARCHAR(100) NOT NULL,
5      PRIMARY KEY(Treatment_type)
6  );
7
```

Figure 43:Create Treatment_type

```
1  INSERT INTO Treatment_type (Treatment_type, Medications, Dosage)
2  VALUES
3  ('Medication', 'Amlodipine', '5mg once daily'),
4  ('Lifestyle Management and Medication', 'Metformin', '500mg twice daily'),
5  ('Surgery', 'Salbutamol', '2 puffs as needed before surgery'),
6  ('Pain Management', 'Ibuprofen', '200mg every 6 hours as needed'),
7  ('Physical Therapy and Medication', 'Paracetamol', '500mg every 4-6 hours')
8  ;
```

Figure 44:Insert into Treatment_type

	#	Name	Type	Collation	Attributes	Null	Default
☐	1	**Treatment_type** 🔑	varchar(100)	utf8mb4_general_ci		No	*None*
☐	2	**Medications**	varchar(255)	utf8mb4_general_ci		No	*None*
☐	3	**Dosage**	varchar(100)	utf8mb4_general_ci		No	*None*

Figure 45:Treatment_type structure

←T→	Treatment_type	Medications	Dosage
☐ Edit Copy Delete	Lifestyle Management and Medication	Metformin	500mg twice daily
☐ Edit Copy Delete	Medication	Amlodipine	5mg once daily
☐ Edit Copy Delete	Pain Management	Ibuprofen	200mg every 6 hours as needed
☐ Edit Copy Delete	Physical Therapy and Medication	Paracetamol	500mg every 4-6 hours
☐ Edit Copy Delete	Surgery	Salbutamol	2 puffs as needed before surgery

Figure 46:Treatment_type table

12. Room table
Room(Room_id,Room_type,Ward_id)

#Creating Room table

```
CREATE TABLE Room (
        Room_id INT(10) NOT NULL,
        Room_type VARCHAR(50) NOT NULL,
        Ward_id INT(10) NOT NULL,
        PRIMARY KEY (Room_id) ,
        FOREIGN KEY (Ward_id) REFERENCES Ward(Ward_id)
 );
```

#Inserting values to Room

```
INSERT INTO Room (Room_id, Room_type, Ward_id)
VALUES
(1, 'Standard', 1),
(2, 'Deluxe', 2),
(3, 'Standard', 3),
(4, 'Suite', 4),
(5, 'Deluxe', 5)
;
```

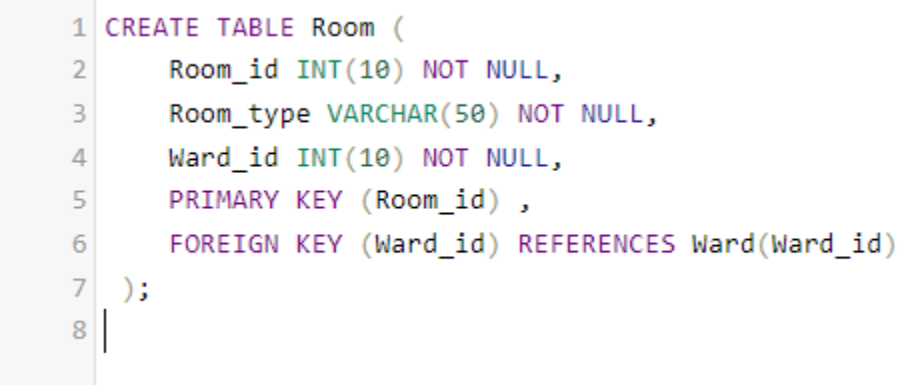


Figure 47:Create room

```sql
INSERT INTO Room (Room_id, Room_type, Ward_id)
VALUES
(1, 'Standard', 1),
(2, 'Deluxe', 2),
(3, 'Standard', 3),
(4, 'Suite', 4),
(5, 'Deluxe', 5)
;
```

Figure 48:Insert into Room

	#	Name	Type	Collation	Attributes	Null	Default
☐	1	**Room_id** 🔑	int(10)			No	*None*
☐	2	**Room_type**	varchar(50)	utf8mb4_general_ci		No	*None*
☐	3	**Ward_id**	int(10)			No	*None*

Figure 49:Room structure

←T→			Room_id	Room_type	Ward_id
☐	Edit	Copy Delete	1	Standard	1
☐	Edit	Copy Delete	2	Deluxe	2
☐	Edit	Copy Delete	3	Standard	3
☐	Edit	Copy Delete	4	Suite	4
☐	Edit	Copy Delete	5	Deluxe	5

Figure 50:Room table

13. Ward_allocation table
Ward_allocation(Ward_id,Floor)

#Creating Ward_allocation table
```
CREATE TABLE Ward_allocation (
   Ward_id INT(10) NOT NULL,
   Floor INT NOT NULL,
   PRIMARY KEY(Ward_id)
);
```

#Inserting values to Ward_allocation
```
INSERT INTO Ward_allocation (Ward_id, Floor)
VALUES
(1, 01),
(2, 02),
(3, 02),
(4, 03),
(5, 04)
;
```

```
1  CREATE TABLE Ward_allocation (
2      Ward_id INT(10) NOT NULL,
3      Floor INT NOT NULL,
4      PRIMARY KEY(Ward_id)
5  );
6
```

Figure 51:Create ward_allocation

```
1  INSERT INTO Ward_allocation (Ward_id, Floor)
2  VALUES
3  (1, 01),
4  (2, 02),
5  (3, 02),
6  (4, 03),
7  (5, 04)
8  ;
9
```

Figure 52:Insert into ward_allocation

	#	Name	Type	Collation	Attributes	Null	Default
☐	1	**Ward_id**	int(10)			No	*None*
☐	2	**Floor**	int(11)			No	*None*

Figure 53:ward_allocation strcuture

	Ward_id	Floor
☐ Edit Copy Delete	1	1
☐ Edit Copy Delete	2	2
☐ Edit Copy Delete	3	2
☐ Edit Copy Delete	4	3
☐ Edit Copy Delete	5	4

Figure 54:ward_allocation table

14. In_patient table
In_patient(<u>In_patient_id</u>,FirstName,LastName,Date_of_birth,Guardian)

#Creating In_patient table

```
CREATE TABLE In_patient (
        In_patient_id INT(10) NOT NULL,
        FirstName VARCHAR(50) NOT NULL,
        LastName VARCHAR(50) NOT NULL,
        Date_of_birth DATE NOT NULL,
        Guardian VARCHAR(50) NOT NULL,
        PRIMARY KEY(In_patient_id)
);
```

#Inserting values to In_patient

```
INSERT INTO In_patient (In_patient_id, FirstName, LastName, Date_of_birth, Guardian) VALUES
(1, 'John', 'Walker', '1990-05-15', 'Jane Walker'),
(2, 'Emily', 'White', '2005-08-22', 'Robert White'),
(3, 'Michael', 'Richards', '1988-12-11', 'Sarah Richards'),
(4, 'Sophia', 'Taylor', '2010-03-29', 'William Taylor'),
(5, 'Daniel', 'Clark', '2000-07-17', 'Ann clark')
;
```

```
1 CREATE TABLE In_patient (
2 In_patient_id INT(10) NOT NULL,
3 FirstName VARCHAR(50) NOT NULL,
4 LastName VARCHAR(50) NOT NULL,
5 Date_of_birth DATE NOT NULL,
6 Guardian VARCHAR(50) NOT NULL,
7 PRIMARY KEY(In_patient_id)
8 );
9
```

Figure 55:Create In_patient

```sql
INSERT INTO In_patient (In_patient_id, FirstName, LastName, Date_of_birth, Guardian)
VALUES
(1, 'John', 'Walker', '1990-05-15', 'Jane Walker'),
(2, 'Emily', 'White', '2005-08-22', 'Robert White'),
(3, 'Michael', 'Richards', '1988-12-11', 'Sarah Richards'),
(4, 'Sophia', 'Taylor', '2010-03-29', 'William Taylor'),
(5, 'Daniel', 'Clark', '2000-07-17', 'Ann clark')
;
```

Figure 56:Insert into In_patient

	#	Name	Type	Collation	Attributes	Null	Default
☐	1	**In_patient_id** 🔑	int(10)			No	*None*
☐	2	**FirstName**	varchar(50)	utf8mb4_general_ci		No	*None*
☐	3	**LastName**	varchar(50)	utf8mb4_general_ci		No	*None*
☐	4	**Date_of_birth**	date			No	*None*
☐	5	**Guardian**	varchar(50)	utf8mb4_general_ci		No	*None*

Figure 57:In_patient structure

←T→	In_patient_id	FirstName	LastName	Date_of_birth	Guardian
☐ Edit Copy Delete	1	John	Walker	1990-05-15	Jane Walker
☐ Edit Copy Delete	2	Emily	White	2005-08-22	Robert White
☐ Edit Copy Delete	3	Michael	Richards	1988-12-11	Sarah Richards
☐ Edit Copy Delete	4	Sophia	Taylor	2010-03-29	William Taylor
☐ Edit Copy Delete	5	Daniel	Clark	2000-07-17	Ann clark

Figure 58:In_patient table

15. In_patient_admission_details table

In_patient_admission_details(In_patient_id,AdmissionDate&Time,Discharge Date&Time,
Room_id)

#Creating In_patient_admission_details table

```
CREATE TABLE In_patient_admission_details (
   In_patient_id INT(10) NOT NULL,
   AdmissionDateTime DATETIME NOT NULL,
   DischargeDateTime DATETIME,
   Room_id INT(10) NOT NULL,
   PRIMARY KEY (In_patient_id),
   FOREIGN KEY (Room_id) REFERENCES Room(Room_id)
);
```

#Inserting values to In_patient_admission_details

```
INSERT INTO In_patient_admission_details (In_patient_id, AdmissionDateTime, DischargeDateTime, Room_id)
VALUES
(101, '2024-12-01 10:00:00', '2024-12-05 14:00:00', 1),
(102, '2024-12-02 09:30:00', '2024-12-06 15:30:00', 2),
(103, '2024-12-03 11:00:00', NULL, 3),
(104, '2024-12-04 08:15:00', '2024-12-07 13:00:00', 4),
(105, '2024-12-05 10:45:00', '2024-12-10 12:00:00', 5);
```

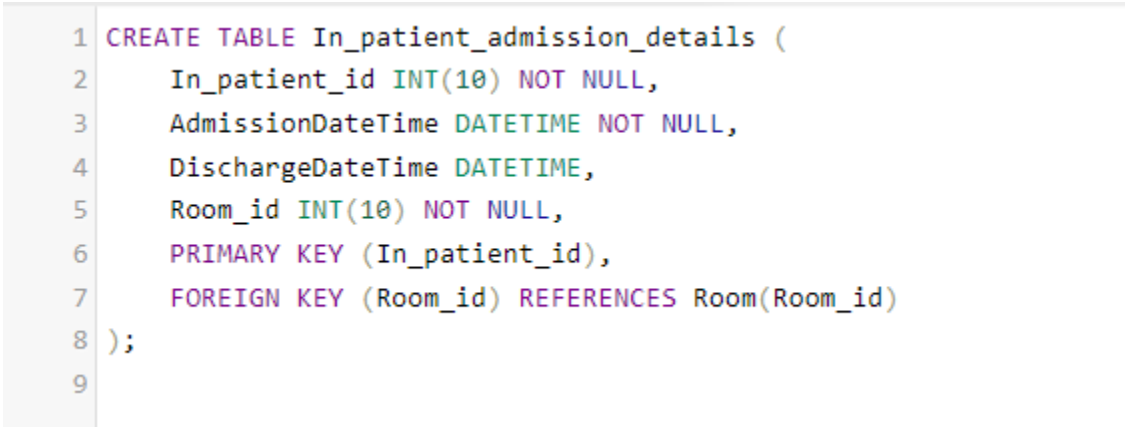


Figure 59:Create In_patient_admission_details

```sql
INSERT INTO In_patient_admission_details (In_patient_id, AdmissionDateTime, DischargeDateTime,
Room_id)
VALUES
(101, '2024-12-01 10:00:00', '2024-12-05 14:00:00', 1),
(102, '2024-12-02 09:30:00', '2024-12-06 15:30:00', 2),
(103, '2024-12-03 11:00:00', NULL, 3),
(104, '2024-12-04 08:15:00', '2024-12-07 13:00:00', 4),
(105, '2024-12-05 10:45:00', '2024-12-10 12:00:00', 5);

```

Figure 60:insert into In_patient_admission_details

	#	Name	Type	Collation	Attributes	Null	Default
☐	1	**In_patient_id** 🔑	int(10)			No	*None*
☐	2	**AdmissionDateTime**	datetime			No	*None*
☐	3	**DischargeDateTime**	datetime			Yes	*NULL*
☐	4	**Room_id** 🔗	int(10)			No	*None*

Figure 61:In_patient_admission_details structure

←T→ ▾	In_patient_id	AdmissionDateTime	DischargeDateTime	Room_id
☐ Edit Copy Delete	101	2024-12-01 10:00:00	2024-12-05 14:00:00	1
☐ Edit Copy Delete	102	2024-12-02 09:30:00	2024-12-06 15:30:00	2
☐ Edit Copy Delete	103	2024-12-03 11:00:00	*NULL*	3
☐ Edit Copy Delete	104	2024-12-04 08:15:00	2024-12-07 13:00:00	4
☐ Edit Copy Delete	105	2024-12-05 10:45:00	2024-12-10 12:00:00	5

Figure 62:In_patient_admission_details table

16. Payment table
Payment=(Payment_id,Payment_method,Amount,Out_patient_id,In_patient_id)

```
#Create Payment table
CREATE TABLE Payment (
   Payment_id INT(11) NOT NULL,
   Payment_method VARCHAR(50) NOT NULL,
   Amount DECIMAL(10, 2) NOT NULL,
   Out_patient_id INT(11),
   In_patient_id INT(11),
   CONSTRAINT check_patient CHECK (
      (Out_patient_id IS NOT NULL AND In_patient_id IS NULL) OR
      (In_patient_id IS NOT NULL AND Out_patient_id IS NULL)
   ),
   FOREIGN KEY (Out_patient_id) REFERENCES Out_patient(Out_patient_id),
   FOREIGN KEY (In_patient_id) REFERENCES In_patient(In_patient_id)
);


#Inserting values to Payment
INSERT INTO Payment (Payment_method, Amount, Out_patient_id, In_patient_id)
VALUES
('Cash', 500.00, NULL, 1),
('Credit Card', 1200.00, 2, NULL),
('Insurance', 750.50, NULL,2),
('Cash', 300.00, 3, NULL),
('Online Transfer', 1000.00, NULL, 3);
```

```sql
CREATE TABLE Payment (
    Payment_id INT(11) NOT NULL,
    Payment_method VARCHAR(50) NOT NULL,
    Amount DECIMAL(10, 2) NOT NULL,
    Out_patient_id INT(11),
    In_patient_id INT(11),
    CONSTRAINT check_patient CHECK (
        (Out_patient_id IS NOT NULL AND In_patient_id IS NULL) OR
        (In_patient_id IS NOT NULL AND Out_patient_id IS NULL)
    ),
    FOREIGN KEY (Out_patient_id) REFERENCES Out_patient(Out_patient_id),
    FOREIGN KEY (In_patient_id) REFERENCES In_patient(In_patient_id)
```

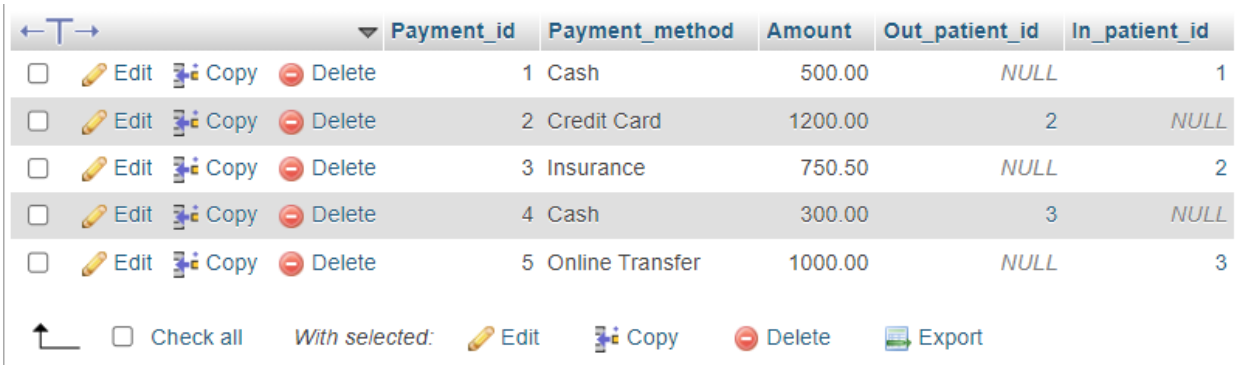
Figure 63:Create payment

```sql
INSERT INTO Payment (Payment_method, Amount, Out_patient_id, In_patient_id)
VALUES
('Cash', 500.00, NULL, 1),
('Credit Card', 1200.00, 2, NULL),
('Insurance', 750.50, NULL,2),
('Cash', 300.00, 3, NULL),
('Online Transfer', 1000.00, NULL, 3);
```

Figure 64:Insert into Payment

	#	Name	Type	Collation	Attributes	Null	Defaul
☐	1	**Payment_id** 🔑	int(11)			No	*None*
☐	2	**Payment_method**	varchar(50)	utf8mb4_general_ci		No	*None*
☐	3	**Amount**	decimal(10,2)			No	*None*
☐	4	**Out_patient_id** 🔗	int(11)			Yes	*NULL*
☐	5	**In_patient_id** 🔗	int(11)			Yes	*NULL*

Figure 65:paymnet structure



	Payment_id	Payment_method	Amount	Out_patient_id	In_patient_id
Edit Copy Delete	1	Cash	500.00	NULL	1
Edit Copy Delete	2	Credit Card	1200.00	2	NULL
Edit Copy Delete	3	Insurance	750.50	NULL	2
Edit Copy Delete	4	Cash	300.00	3	NULL
Edit Copy Delete	5	Online Transfer	1000.00	NULL	3

Check all *With selected:* Edit Copy Delete Export

Figure 66:payment table

17. Doctor_in_patient table
Doctor_in_patient=((Doctor_id,In_patient_id))

#Create Doctor_in_patient table
CREATE TABLE Doctor_in_patient (
Doctor_id INT(10) NOT NULL,
In_patient_id INT(10) NOT NULL,
PRIMARY KEY (Doctor_id, In_patient_id),
FOREIGN KEY (Doctor_id) REFERENCES In_house_doctor(Doctor_id),
FOREIGN KEY (Doctor_id) REFERENCES In_house_doctor(Doctor_id),
FOREIGN KEY (In_patient_id) REFERENCES In_patient(In_patient_id)
);

#Inserting values to Doctor_in_patient
INSERT INTO Doctor_in_patient (Doctor_id, In_patient_id)
VALUES
(1, 1),
(2, 2),
(1, 3),
(3, 4),
(2, 5);

```
CREATE TABLE Doctor_in_patient (
    Doctor_id INT(10) NOT NULL,
    In_patient_id INT(10) NOT NULL,
    PRIMARY KEY (Doctor_id, In_patient_id),
    FOREIGN KEY (Doctor_id) REFERENCES In_house_doctor(Doctor_id),
    FOREIGN KEY (Doctor_id) REFERENCES In_house_doctor(Doctor_id),
    FOREIGN KEY (In_patient_id) REFERENCES In_patient(In_patient_id)
);
```

Figure 67:Create Doctor_in_patient

```sql
INSERT INTO Doctor_in_patient (Doctor_id, In_patient_id)
VALUES
(1, 1),
(2, 2),
(1, 3),
(3, 4),
(2, 5);
```

Figure 68:Insert into Doctor_in_patient

	#	Name	Type	Collation	Attributes	Null	Default
☐	1	**Doctor_id** 🔑	int(10)			No	*None*
☐	2	**In_patient_id** 🔑 🔗	int(10)			No	*None*

Figure 69:Doctor_in_patient structure

←T→	Doctor_id	In_patient_id
☐ Edit Copy Delete	1	1
☐ Edit Copy Delete	1	3
☐ Edit Copy Delete	2	2
☐ Edit Copy Delete	2	5
☐ Edit Copy Delete	3	4

Figure 70:Doctor_in_patient table

18. Doctor_treatment table

Doctor_treatment=((Doctor_id,Treatment_id))

#Create Doctor_treatment table
CREATE TABLE Doctor_treatment (
Doctor_id INT(10) NOT NULL,
Treatment_id INT(6) NOT NULL,
PRIMARY KEY (Doctor_id, Treatment_id),
FOREIGN KEY (Doctor_id) REFERENCES In_house_doctor(Doctor_id),
FOREIGN KEY (Doctor_id) REFERENCES Visiting_doctor(Doctor_id),
FOREIGN KEY (Treatment_id) REFERENCES Treatment(Treatment_id)
);

#Inserting values to Doctor_treatment
INSERT INTO Doctor_treatment (Doctor_id, Treatment_id)
VALUES
(1, 1),
(2, 2),
(1, 3),
(3, 1),
(2, 4)
;

```
1 CREATE TABLE Doctor_treatment (
2 Doctor_id INT(10) NOT NULL,
3 Treatment_id INT(6) NOT NULL,
4 PRIMARY KEY (Doctor_id, Treatment_id),
5 FOREIGN KEY (Doctor_id) REFERENCES In_house_doctor(Doctor_id),
6 FOREIGN KEY (Doctor_id) REFERENCES Visiting_doctor(Doctor_id),
7 FOREIGN KEY (Treatment_id) REFERENCES Treatment(Treatment_id)
8 );
9
```

Figure 71:Create Doctor_treatment

```sql
INSERT INTO Doctor_treatment (Doctor_id, Treatment_id)
VALUES
(1, 1),
(2, 2),
(1, 3),
(3, 1),
(2, 4)
;
```

Figure 72:Insert into Doctor_treatment

	#	Name	Type	Collation	Attributes	Null	Default
☐	1	**Doctor_id** 🔑	int(10)			No	*None*
☐	2	**Treatment_id** 🔑 🔗	int(6)			No	*None*

Figure 73:Doctor_treatment structure

←T→		Doctor_id	Treatment_id
☐	Edit Copy Delete	1	1
☐	Edit Copy Delete	1	3
☐	Edit Copy Delete	2	2
☐	Edit Copy Delete	2	4
☐	Edit Copy Delete	3	1

Figure 74:Doctor_treatment table

19. In_patient_treatment table
In_patient_treatment=((In_patient_id,Treatment_id))

#Create In_patient_treatment table
CREATE TABLE In_patient_treatment (
In_patient_id INT(10) NOT NULL,
Treatment_id INT(6) NOT NULL,
PRIMARY KEY (In_patient_id, Treatment_id),
FOREIGN KEY (In_patient_id) REFERENCES In_patient(In_patient_id),
FOREIGN KEY (Treatment_id) REFERENCES Treatment(Treatment_id)
);

#Inserting values to In_patient_treatment
INSERT INTO In_patient_treatment (In_patient_id, Treatment_id)
VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 1),
(5, 4);

```
CREATE TABLE In_patient_treatment (
    In_patient_id INT(10) NOT NULL,
    Treatment_id INT(6) NOT NULL,
    PRIMARY KEY (In_patient_id, Treatment_id),
    FOREIGN KEY (In_patient_id) REFERENCES In_patient(In_patient_id),
    FOREIGN KEY (Treatment_id) REFERENCES Treatment(Treatment_id)
);

```

Figure 75:Create In_patient_treatment

```
INSERT INTO In_patient_treatment (In_patient_id, Treatment_id)
VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 1),
(5, 4);
```

Figure 76:Insert into In_patient_treatment

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null
Edit Rename Drop	PRIMARY	BTREE	Yes	No	In_patient_id	5	A	No
					Treatment_id	5	A	No
Edit Rename Drop	Treatment_id	BTREE	No	No	Treatment_id	5	A	No

Figure 77:In_patient_treatment structure

	In_patient_id	Treatment_id
Edit Copy Delete	1	1
Edit Copy Delete	2	2
Edit Copy Delete	3	3
Edit Copy Delete	4	1
Edit Copy Delete	5	4

Figure 78:In_patient_treatment table

20. Schedule table
Schedule=((Doctor_id, Out_patient_id ,Appointment_id))

#Create Shedule table
CREATE TABLE Schedule (
Doctor_id (10)INT NOT NULL,
Out_patient_id INT(10) NOT NULL,
Appointment_id INT(11) NOT NULL,
PRIMARY KEY (Doctor_id, Out_patient_id, Appointment_id),
FOREIGN KEY (Doctor_id) REFERENCES In_house_doctor(Doctor_id),
FOREIGN KEY (Doctor_id) REFERENCES Visiting_doctor(Doctor_id),
FOREIGN KEY (Out_patient_id) REFERENCES
Out_patient(Out_patient_id),
FOREIGN KEY (Appointment_id) REFERENCES
Appointment(Appointment_id)
);

#Inserting values to Shedule
INSERT INTO Schedule (Doctor_id, Out_patient_id, Appointment_id)
VALUES
(1, 1, 101),
(2, 2, 102),
(3, 3, 103),
(1, 4, 104),
(2, 5, 105);

```
1  CREATE TABLE Schedule (
2  Doctor_id INT(10) NOT NULL,
3  Out_patient_id INT(10) NOT NULL,
4  Appointment_id INT(11) NOT NULL,
5  PRIMARY KEY (Doctor_id, Out_patient_id, Appointment_id),
6  FOREIGN KEY (Doctor_id) REFERENCES In_house_doctor(Doctor_id),
7  FOREIGN KEY (Doctor_id) REFERENCES Visiting_doctor(Doctor_id),
8  FOREIGN KEY (Out_patient_id) REFERENCES Out_patient(Out_patient_id),
9  FOREIGN KEY (Appointment_id) REFERENCES Appointment(Appointment_id)
10 );
11
```

Figure 79:Create schedule

```sql
INSERT INTO Schedule (Doctor_id, Out_patient_id, Appointment_id)
VALUES
(1, 1, 101),
(2, 2, 102),
(3, 3, 103),
(1, 4, 104),
(2, 5, 105);
```

Figure 80:Insert into Schedule

#	Name	Type	Collation	Attributes	Null	Default
1	**Doctor_id**	int(10)			No	*None*
2	**Out_patient_id**	int(10)			No	*None*
3	**Appointment_id**	int(11)			No	*None*

Figure 81:Schedule structure

	Doctor_id	Out_patient_id	Appointment_id
Edit Copy Delete	1	1	101
Edit Copy Delete	1	4	104
Edit Copy Delete	2	2	102
Edit Copy Delete	2	5	105
Edit Copy Delete	3	3	103

Figure 82:Schedule table

Database Diagram

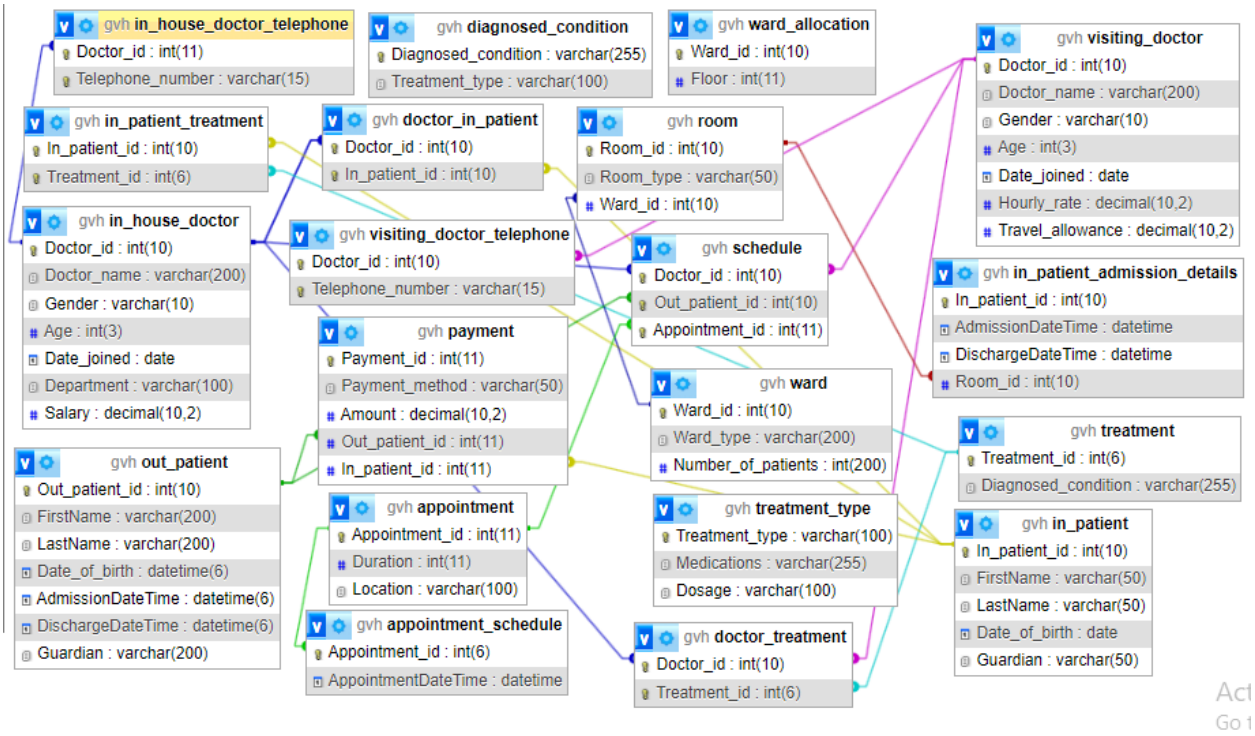


Figure 83:Databse Diagram

Section 5-Data Manipulation with SQL

Qusetion1

```
SELECT Doctor_name, Specialization
FROM (
    SELECT Doctor_name, Specialization, Date_joined
    FROM In_house_doctor
    WHERE Specialization IN ('Cardiology', 'Neurology') AND Date_joined < '2022-01-01'

    UNION ALL

    SELECT Doctor_name, Specialization, Date_joined
    FROM Visiting_doctor
    WHERE Specialization IN ('Cardiology', 'Neurology') AND Date_joined < '2022-01-01'
) AS doctors
ORDER BY Date_joined DESC;
```

```
1  SELECT Doctor_name, Specialization
2  FROM (
3      SELECT Doctor_name, Specialization, Date_joined
4      FROM In_house_doctor
5      WHERE Specialization IN ('Cardiology', 'Neurology') AND Date_joined < '2022-01-01'
6
7      UNION ALL
8
9      SELECT Doctor_name, Specialization, Date_joined
10     FROM Visiting_doctor
11     WHERE Specialization IN ('Cardiology', 'Neurology') AND Date_joined < '2022-01-01'
12 ) AS doctors
13 ORDER BY Date_joined DESC;
14
15
```

Doctor_name	Specialization
Dr. Robert Smith	Neurology
Dr. Emily Johnson	neurology
Dr. Alice Brown	Cardiology
Dr. John Smith	Cardiology
Dr. Karen Johnson	Cardiology

Question 2

SELECT w.Ward_type, w.Number_of_patients - COUNT(r.Room_id) AS
Remaining_Patient_Availability
FROM Ward w
JOIN Room r ON w.Ward_id = r.Ward_id
WHERE r.Room_type = 'Deluxe'
GROUP BY w.Ward_id, w.Ward_type, w.Number_of_patients
HAVING w.Number_of_patients > 5;

```sql
SELECT w.Ward_type, w.Number_of_patients - COUNT(r.Room_id) AS Remaining_Patient_Availability
FROM Ward w
JOIN Room r ON w.Ward_id = r.Ward_id
WHERE r.Room_type = 'Deluxe'
GROUP BY w.Ward_id, w.Ward_type, w.Number_of_patients
HAVING w.Number_of_patients > 5;
```

Ward_type	Remaining_Patient_Availability
Maternity	11

Question 3

```
SELECT SUM(p.Amount) AS Total_Cash_Received
FROM Payment p
JOIN Schedule s ON p.Out_patient_id = s.Out_patient_id
JOIN Appointment_schedule ap ON s.Appointment_id = ap.Appointment_id
JOIN In_house_doctor d ON s.Doctor_id = d.Doctor_id
WHERE d.Specialization = 'Pediatric'
  AND d.Surname IN ('Peterson', 'Fedrick')
  AND p.Payment_method = 'Cash'
  AND ap.AppointmentDateTime >= DATE_SUB(CURDATE(), INTERVAL 10 DAY);
```


```
SELECT SUM(p.Amount) AS Total_Cash_Received
FROM Payment p
JOIN Schedule s ON p.Out_patient_id = s.Out_patient_id
JOIN Appointment_schedule ap ON s.Appointment_id = ap.Appointment_id
JOIN In_house_doctor d ON s.Doctor_id = d.Doctor_id
WHERE d.Specialization = 'Pediatric'
  AND d.Surname IN ('Peterson', 'Fedrick')
  AND p.Payment_method = 'Cash'
  AND ap.AppointmentDateTime >= DATE_SUB(CURDATE(), INTERVAL 10 DAY);
```