



DATABASE PROJECT

GROUP: 2 AMS-A





HOSPITAL MANAGEMENT SYSTEM

*By Group
I3-AMS-A*



LECTURER



Thear Sophal(TP)



Lay Vathna(Course)

OUR TEAM MEMBERS



CHEA PISETH
e20210871



KANG PENGKHEANG
e20210527



Chan Sophara
e20211081



**HAYSAVIN
RONGRAVIDWIN**
e20211502



CHHRAN MOSES
e20210294

1ST WEEK CONTENT

1

About us

2

Introduction

3

Slope(Objective)

4

key
functionalities

5

Entities and its
attributes

6

Conclusion

01

ABOUT US



OUR TEAM

Our team chose the hospital management system for our database project because it is a crucial and widely used system in the healthcare industry. It provides a comprehensive solution for managing various aspects of a hospital, such as patient information, appointments, medical records, billing, and inventory management. By implementing a hospital management system, we aim to streamline and automate the administrative and operational processes within a hospital, improving efficiency, accuracy, and patient care. Additionally, working on a project related to the healthcare domain allows us to gain practical experience and contribute to a field that has a direct impact on people's lives.

02

INTRODUCTION

Our project focuses on designing a database for a Hospital Management System, which is essential in modern healthcare. It involves organizing and managing information about patients, doctors, nurses, appointments, and medical records. By ensuring a well-organized database, we aim to enhance hospital operations and improve patient care. The goal is to demonstrate how a smart database design can significantly impact the smooth functioning of a hospital and enhance the quality of care provided to patients.



SLOPE(OBJECTIVE)

To create a good database for a Hospital Management System, it's important to know what it should do. This includes tasks like registering patients, managing their medical records, and handling billing. These are the key things that make the system work well.

03



KEY FUNCTIONALITIES OF A HOSPITAL MANAGEMENT SYSTEM

THE KEY FUNCTIONALITIES OF A HOSPITAL MANAGEMENT SYSTEM INCLUDE:

- Patient registration and management
- Appointment scheduling
- Medical record management
- Billing and payment processing
- Inventory management

04

ENTITIES AND ITS ATTRIBUTES

**THE FOLLOWING DATA NEEDS TO BE STORED
AND MANAGED BY A HOSPITAL
MANAGEMENT SYSTEM:**

- Patient information (ID, name, address, date of birth, phone Number etc.)
- Doctor (ID, name, specialty, Phone number, etc.)
- Hospital information (ID, name, address, Phone Number)

05

CONCLUSION

In summary, for first presentation, our Hospital Management System database project is a big step to make healthcare work better. We carefully set up a strong database that makes things like patient registration, appointments, medical records, and billing much smoother. The goal is to make sure that important health information is correct, easy to get to, and safe.

WELCOME BACK



2ST WEEK CONTENTS

1

About Our
Improvement

2

Tools

3

Details Entities

4

Entities
Relationship

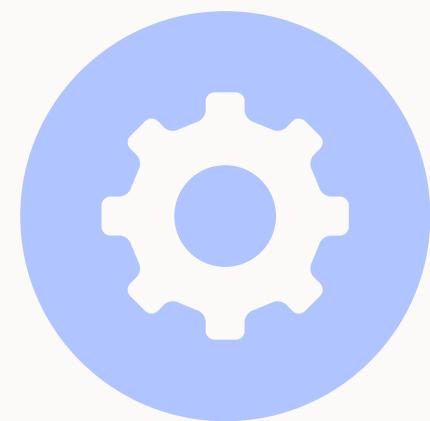
5

Its Attributes

6

Conclusion

ABOUT OUR IMPROVEMENT



ABOUT TOOLS

For my DBMS project, I utilize MySQL Workbench, XAMPP, MAMP, and Python for web app development, leveraging these tools for efficient database management and application integration.



ABOUT ENTITIES

Entities in the Hospital Management System represent key components like patients, staff, medical records, appointments, and inventory, facilitating efficient healthcare operations.

07



ABOUT ER(ENTITY-RELATIONSHIP)

Entity-Relationship (ER) models depict relationships between entities in a database system, illustrating how data entities interact and relate to each other.

08



Tools



MySQL plays a crucial role in a Hospital Management System by serving as the relational database management system (RDBMS) for storing, managing, and retrieving various types of data.



XAMPP can play a role in developing a Hospital Management System by providing a convenient and easy-to-install environment for hosting the necessary components of the system.



MAMP simplifies MySQL management for macOS developers, bundling Apache, MySQL, and PHP for streamlined web app testing and development.



python

Python can be used to build web application for patient to access Hospital management system.



09

Patient

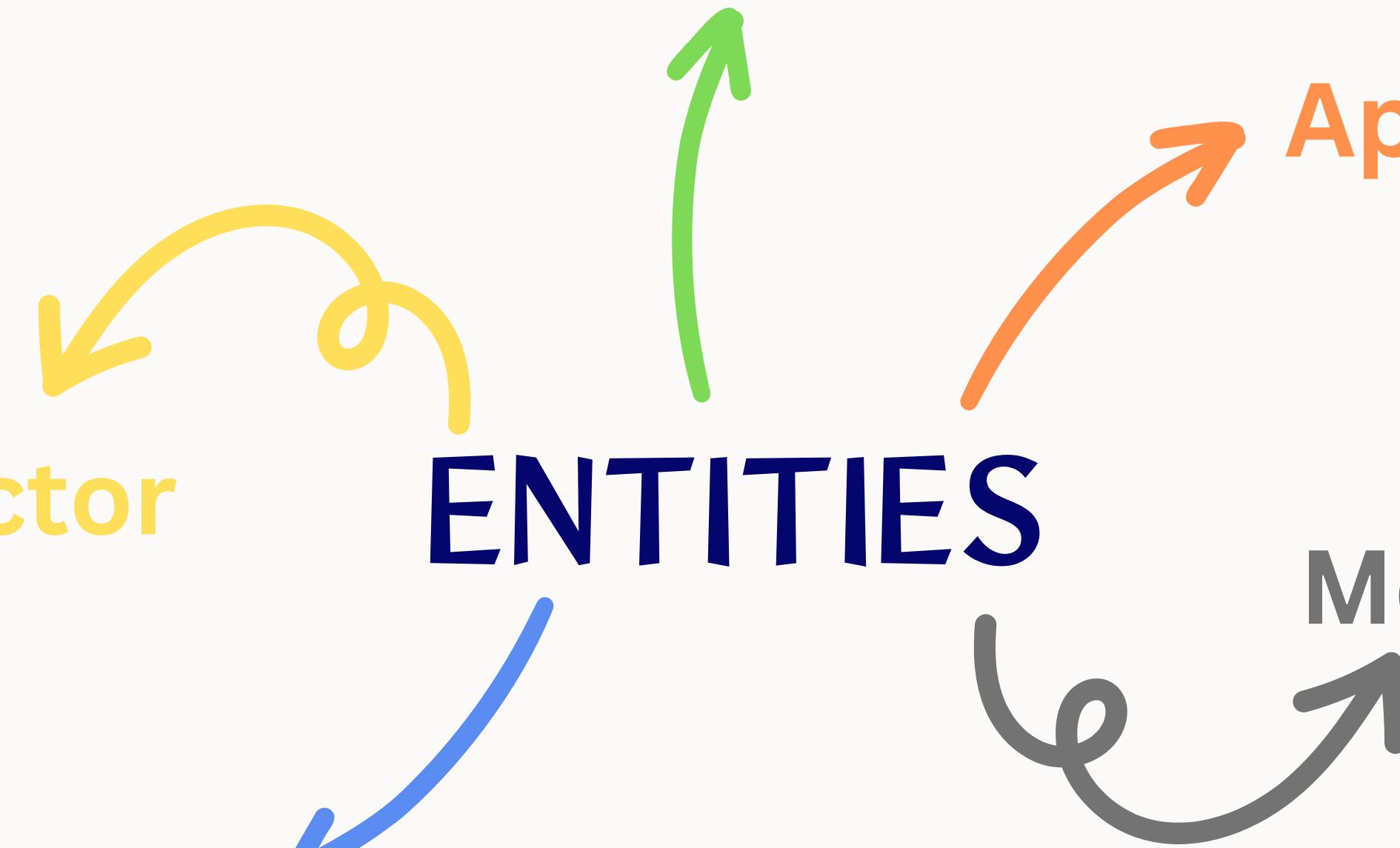
Doctor

Schedule

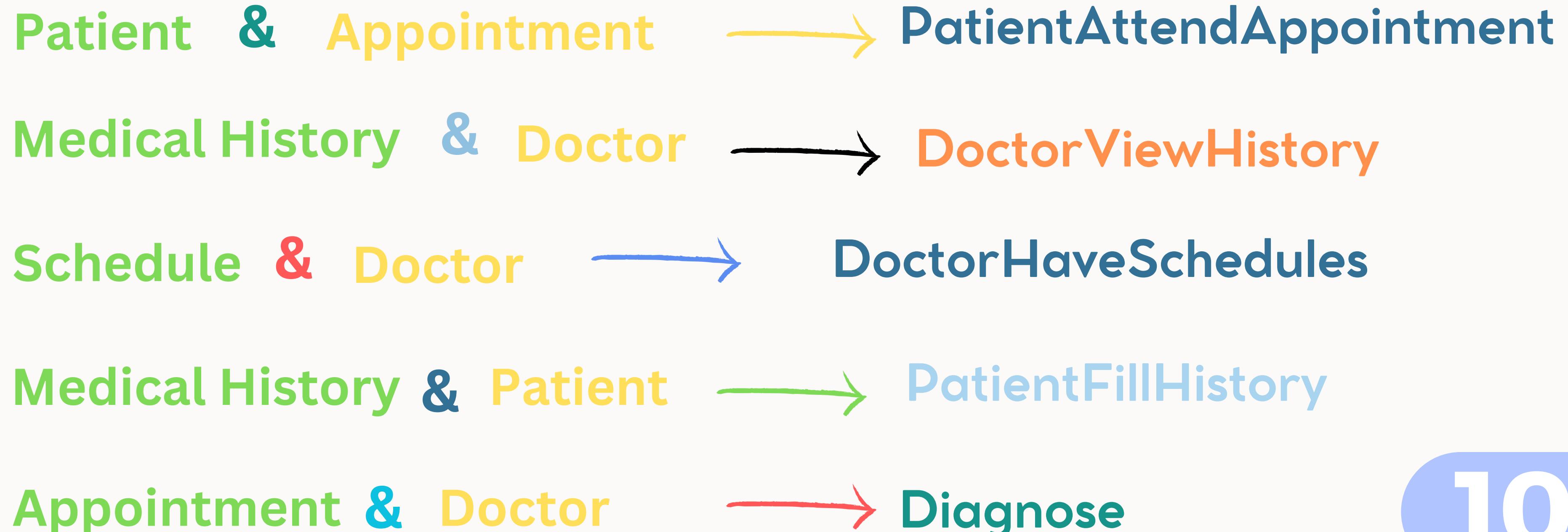
ENTITIES

Appointment

Medical History



ENTITIES RELATIONSHIP



ITS ATTRIBUTES

11

01

Doctor

The "Doctor" entity comprises attributes: "Email," "Gender," "Password," and "Name," facilitating secure and personalized identification and communication within healthcare systems.

02

Patient

The "Patient" entity features attributes: "Email," "Password," "Name," "Address," and "Gender," enabling personalized patient identification and secure access within healthcare systems.

03

Appointment

The "Appointment" entity consists of attributes: "Id," "Date," "Start Time," "End Time," and "Status," essential for scheduling and tracking appointments efficiently.

04

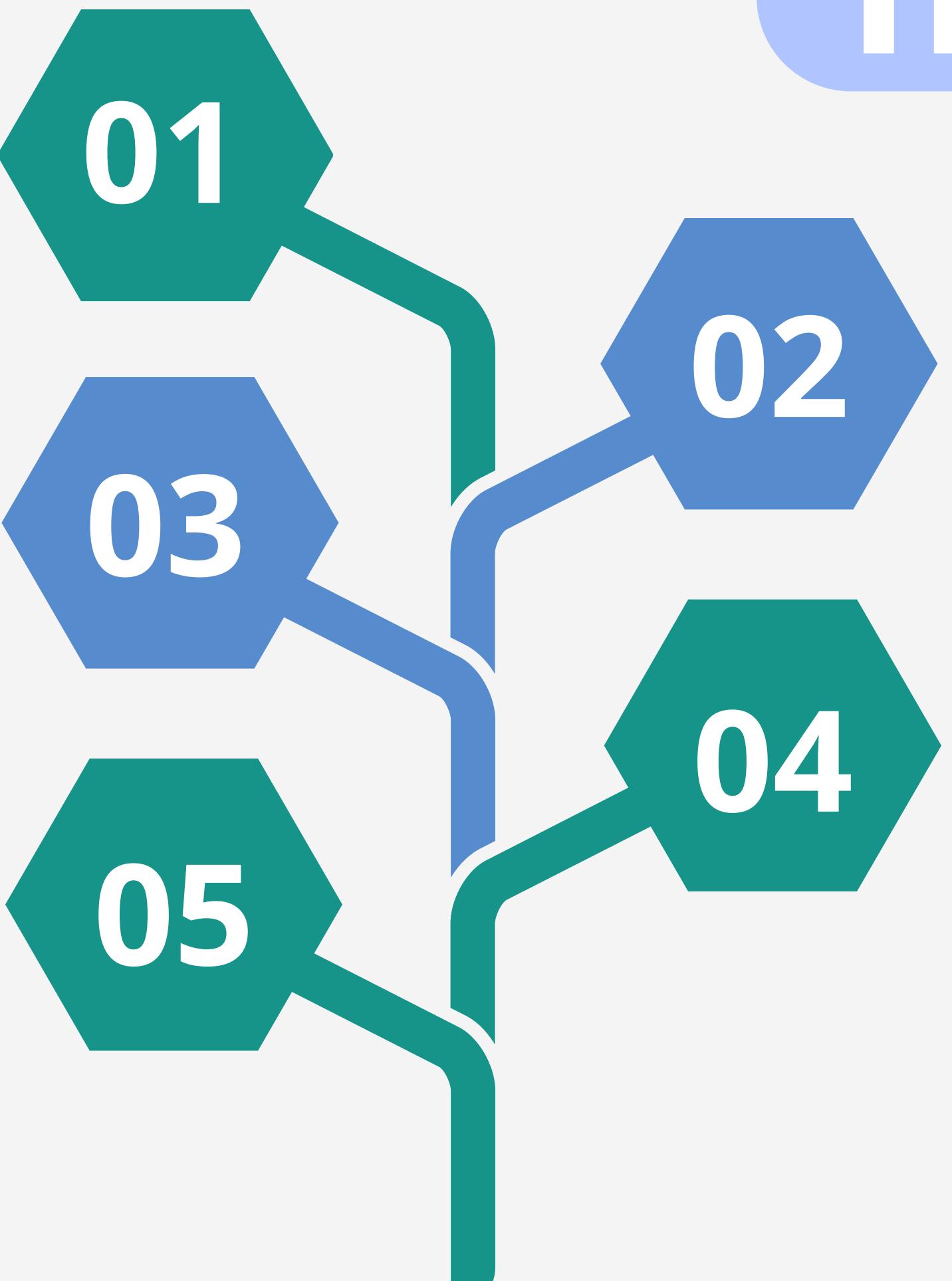
Medical History

The "Medical History" entity contains attributes: "Id," "Date/Time," "Condition," "Surgeries," and "Medication," crucial for documenting patients' health records comprehensively.

05

Schedules

The "Schedules" entity includes attributes: "Id," "Start Time," "End Time," "Break Time," and "Day," facilitating efficient management of time slots for appointments or tasks.



ITS ATTRIBUTES

01

DoctorViewHistory

In the "DoctorViewHistory" entity, attributes include "History" and "Doctor." This entity likely records medical history viewed by doctors, aiding patient care.

02

PatientAttendAppointment

The "PatientAttendAppointment" entity includes attributes: "Patient," "Appointment," "Concerns," and "Symptoms," likely capturing patient details and medical concerns during appointments.

03

Diagnose

The "Diagnose" entity comprises attributes: "Appointment," "Doctor," "Diagnosis," and "Prescription," central to recording medical consultations and treatment plans.

04

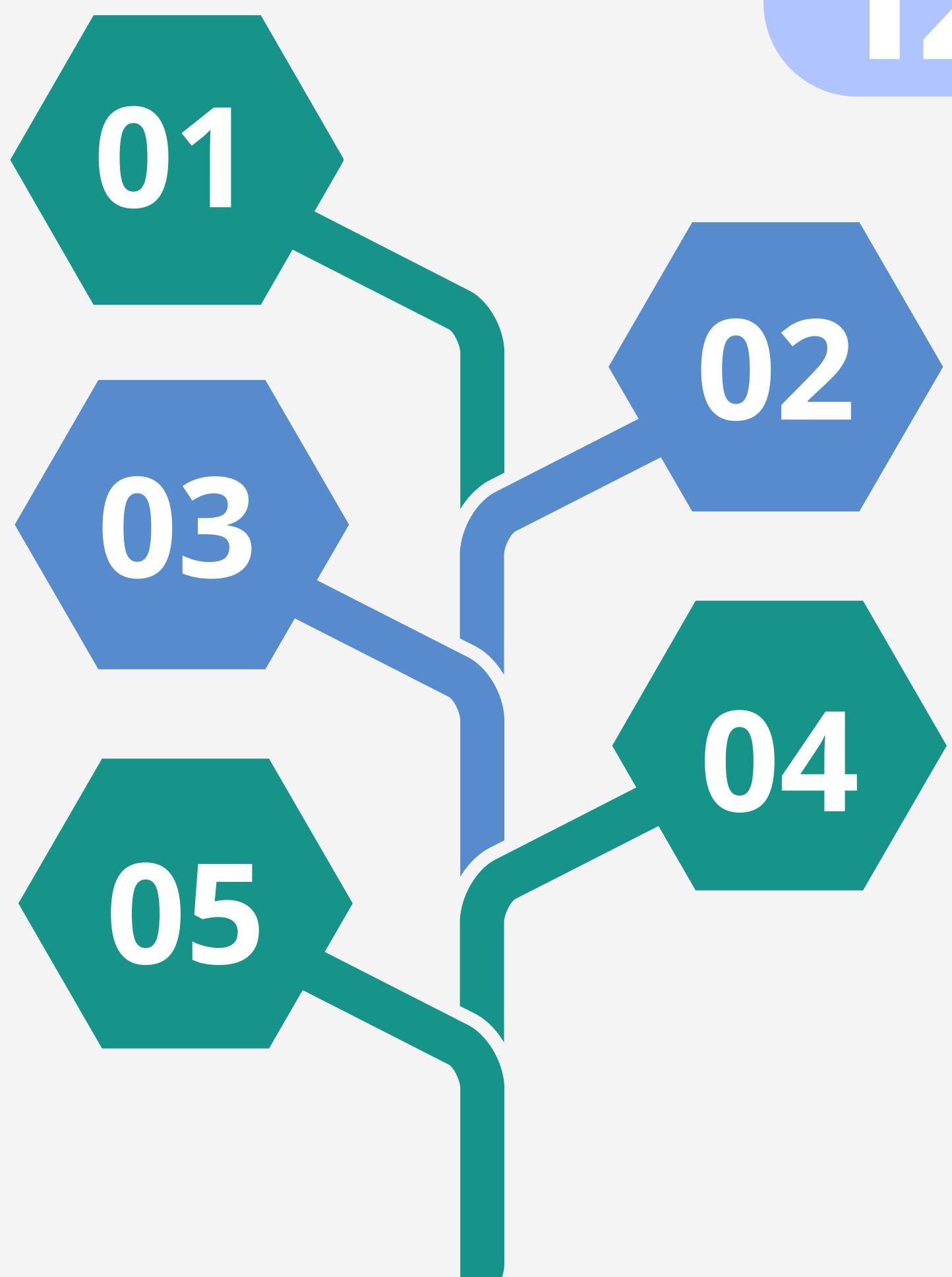
PatientFillHistory

The "PatientFillHistory" entity consists of attributes "Patient" and "History," likely tracking patient-specific medical history information, aiding in comprehensive healthcare management.

05

DoctorHaveSchedules

The "DoctorHaveSchedules" entity includes attributes "Schedule" and "Doctor," likely organizing doctors' schedules for efficient appointment management in healthcare systems.



CONCLUSION

SECOND CONCLUSION

In two weeks of DBMS exploration, I've utilized MySQL Workbench, XAMPP, MAMP, and Python for web apps. Learning ER diagrams and attributes has enhanced my grasp of database structures. This progress check week solidifies my foundation, paving the way for deeper understanding and proficiency in database management.

13





**THANK YOU
VERY MUCH !**



I AM HERE AGAIN

Group: 2 AMS-A

3RD WEEK CONTENTS

1

About Our
Improvement

2

Attributes update

3

Entities update

4

E-R Diagram

5

Relationship
Shemas

6

Conclusion

1.ABOUT OUR IMPROVEMENT

After last week presentation and getting validated by teacher we find some error and new improvement, So for this week we improve our database we are adding:

- 1.Update new attributes to our entities**
- 2.Adding new entities to DATABASE**
- 3.Draw E-R diagram on IO**
- 4.Draw Relationship Schemas on IO**

2. ATTRIBUTES UPDATE

01

Schedule(Entity)

- Availability(doctor & nurse) , so patient can see available doctors and nurses in real time.
- Time Table(Patient) , so Patient can see time available.

02

Doctor(Entity)

- Speciality , to see which doctor is specialize in which illness.

03

Doctor_HaveSchedules

- Request access(Patient), so patient can access doctor schedule.

01

03

02

3.ENTITIES UPDATE

16

01

Change entity name

We are changing entity (Patient_viewAppointment) to (Patient_MakeAppointment) because patient is the one who make appointment and not doctor.

02

Adding new entity(Location)

we are adding new entity to our dataset which related to entity Appointment and Schedule.

Location(entity):

- Building name(attribute)
- Floor(attribute)
- Room number(attribute)

02

01

3. ENTITIES UPDATE(OVER VIEW)

17

Main Entities

Doctor	Patient	Appointments	Schedule	Medical History
Email	Email	ID	ID	ID
Gender	Password	Date	StartTime	Date/Time
Password	Name	Start Time	EndTime	Condition
Name	Address	End Time	BreakTime	Surgeries
Specialty	Gender	Status	Day	Medication
			Availability (doctor & nurse)	
			Time Table	

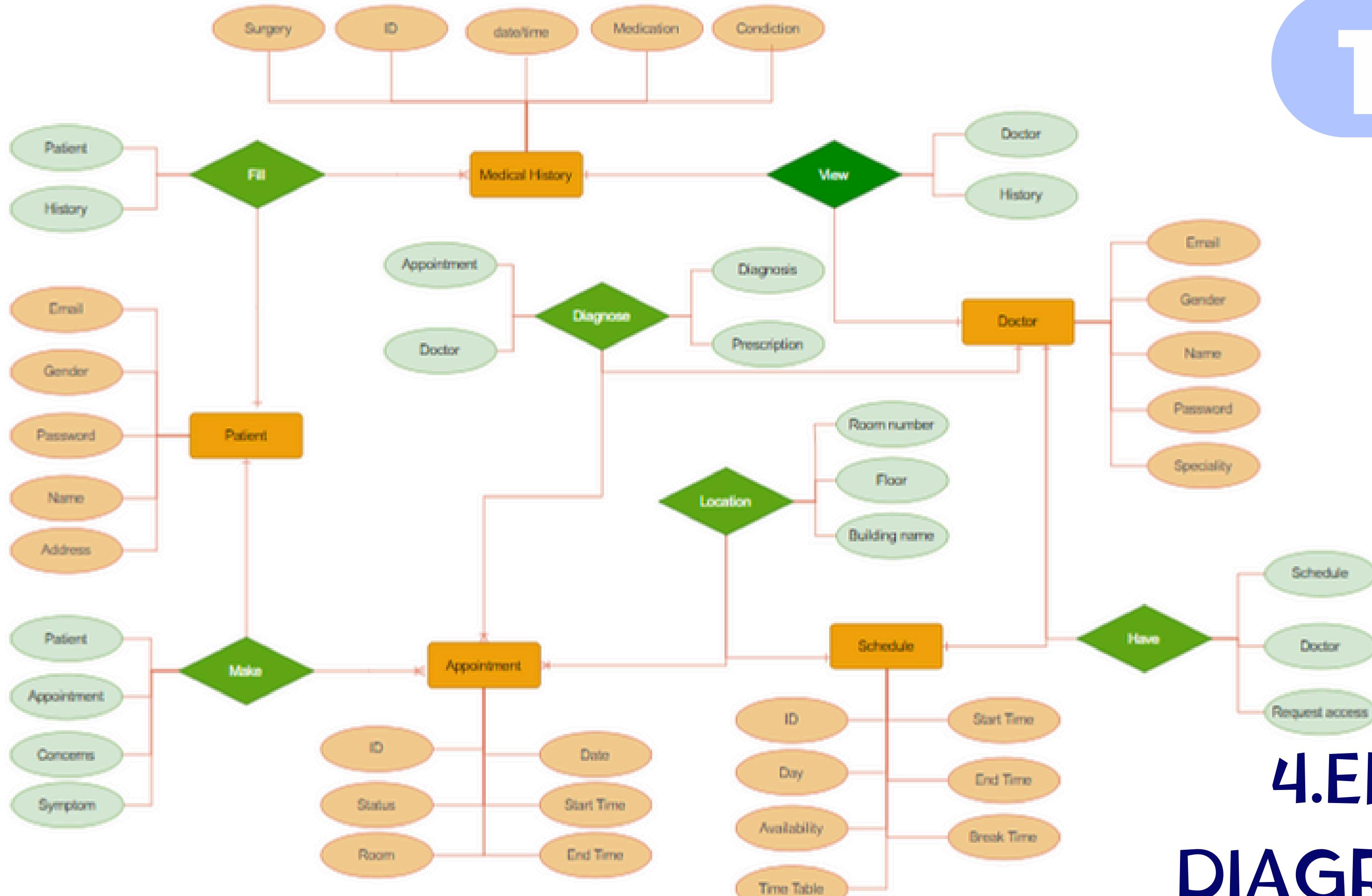
Entities related to Main Entities

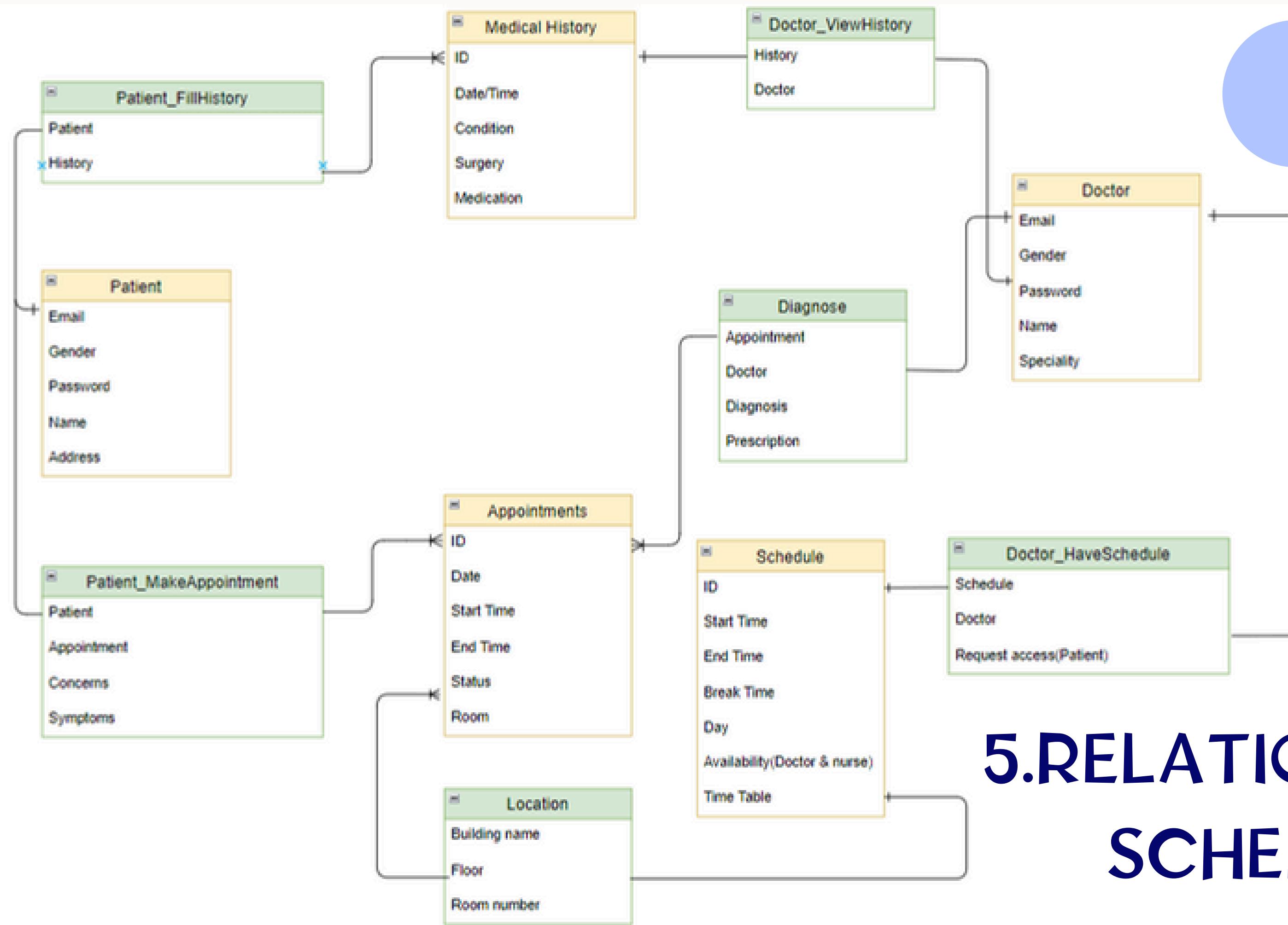
Doctor_ViewHistory	Patient_MakeAppointment	Diagnose	Patient_FillHistory	Doctor_HaveSchedules
History	Patient	Appointmae	Patient	Schedule
Doctor	Appointment	Doctor	History	Doctor
	Concerns		Diagnosis	Request access(Patient)
	Symptoms		Prescription	

New Entities

Location
Building name
Floor
Room number

4.ER DIAGRAM





5.RELATIONSHIP SCHEMA

CONCLUSION

THIRD CONCLUSION

In this third weeks of DBMS exploration, We have updated our entities and attributes and . Learning ER diagrams and Relationship Shemas and attributes has enhanced my grasp of database structures. The progress of this week solidifies my foundation, paving the way for deeper understanding and proficiency in database management.

20





**THANK YOU
VERY MUCH !**

FR PLSS
don't ask
Qestion!



WE'RE BACK

Group: 2 AMS-A

5TH WEEK CONTENTS

1

About Our
Improvement

2

Teacher
Recommendation update

3

Relational and Key

4

Conclusion

1.ABOUT OUR IMPROVEMENT

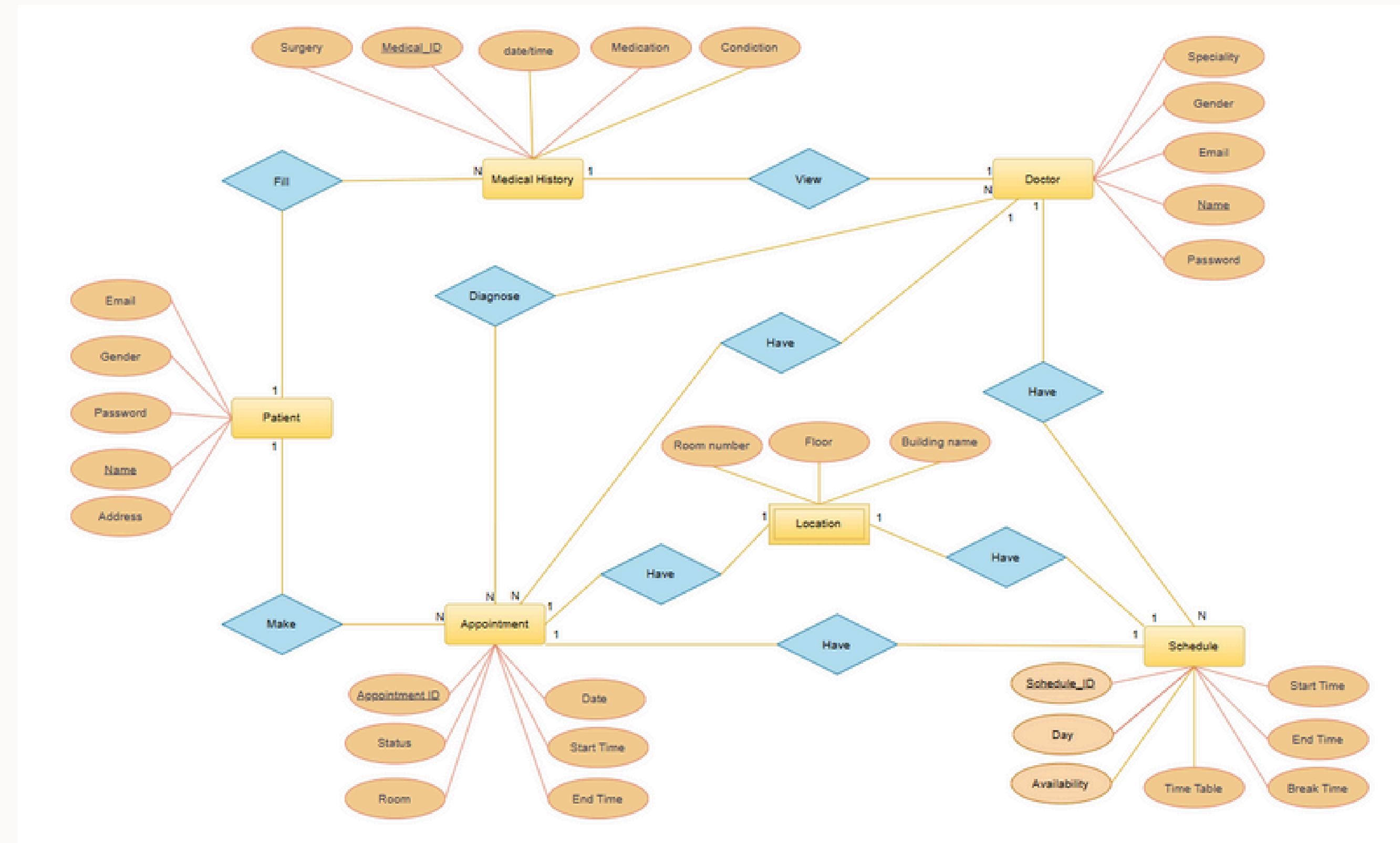
After last week presentation and getting validated by teacher we find some error and new improvement, So for this week we improve our database we are adding:

- 1.Teacher recommendation**
- 2.Relational and Key**

2. TEACHER RECOMMENDATION

1. Change Location(relationship) to Location(entity)
2. if could , make 2 schedule one for doctors and one for patients(but no need to cuz we already combine them)
3. Add relationship(Have) between Appointments(entity) and Schedule(entity).
4. Add relationship(Have) between Appointment(entity) and Location(entity)
5. Add relationship(Have) between doctor(entity) and Appointment(entity)
6. ER Diagram relationship style use CHEN diagram instead
7. ER Diagram relation no need to put attitude in it .
8. Underline Key attributes of each entity And make it unique (Example : ID(attribute) in Medical History (Entity) change to Medical_ID(attribute)
9. Double check spelling
- 10 . Just use straight line to connect attributes to entities when drawing ER diagram so it looks more good

UPDATED ER DIAGRAM



3.RELATION AND KEY

Patient Table					Superkey	{Name}, {Email}, {Name, Email}			
Name	Gender	Address	Email	Password	Candidate Key	{Email} (since it uniquely identifies each patient)			
Jonh	Male	Chba ompov	Jonh@gmail.com		Primary Key	{Email} (chosen from the candidate keys)			
Monika	Female	Chom kadoung	Monika@gmail.com		Foreign Key	None			
Jeck	Male	Streong meanchey	Jeck@gmail.com						

Doctor					Superkey	{Name}, {Email}, {Name, Email}			
Name	Gender	Speciality	Email	Password	Candidate Key	{Email} (since it uniquely identifies each doctor)			
Aurora	Female	Heart	Aurora@gmial.com		Primary Key	{Email} (chosen from the candidate keys)			
Silas Archer	Male	Breathe	SilasArcher@gmail.com		Foreign Key	None			
Leo Bennett	Male	Heart	LeoBennett@gmail.com						

Medical History					Superkey	{Medical ID}			
Medical ID	Date/Time	Surgery	Medication	Condiction	Candidate Key	{Medical ID} (since it uniquely identifies each medical history entry)			
1	2/4/24 8:00	Knee Replacement	Acetaminophen	Appendicitis	Primary Key	{Medical ID} (chosen from the candidate keys)			
2	15/3/24 10:30	Heart Bypass	Oxycodone	Osteoarthritis	Foreign Key	None			
3	20/2/24 14:15	Appendectomy	Aspirin	Coronary Artery Disease					

3.RELATION AND KEY

Appointment							Superkey	(Appointment ID), (Date, Start Time), (Date, Room), (Date, Room, Start Time), (Date, Room, Start Time, End Time)
Appointment ID	Date	Start Time	End Time	Status	Room	Reason	Candidate Key	(Appointment ID), (Date, Start Time) (both uniquely identify each appointment)
1	2/4/24	8:00:00	9:00:00	Scheduled	101		Primary Key	(Appointment ID) (chosen from the candidate keys)
2	2/4/24	9:30:00	10:30:00	Completed	102		Foreign Key	(Room) (references Location(RoomNumber))
3	3/4/24	10:00:00	11:00:00	Scheduled	103			

Location			Superkey	(RoomNumber), (Floor, RoomNumber)
RoomNumber	Floor	Building Name	Candidate Key	(RoomNumber) (since it uniquely identifies each location)
101	1	A	Primary Key	(RoomNumber) (chosen from the candidate keys)
308	3	B	Foreign Key	None
404	4	C		

CONCLUSION

In conclusion, the upgraded hospital management system project showcases enhanced diagrams, incorporates valuable feedback from previous classes' teachers, and meticulously defines relationships and keys, including candidate, primary, and foreign keys. These updates significantly improve the system's functionality and usability, ensuring a streamlined and efficient hospital management experience.

26





**THANK YOU
VERY MUCH !**



WE ARE BACK ONCE AGAIN

Group: 2 AMS-A

THIS WEEK CONTENTS

1

Teacher
recommendation

2

Overview on ER
diagram

3

New update
(ERR on Mysql)

4

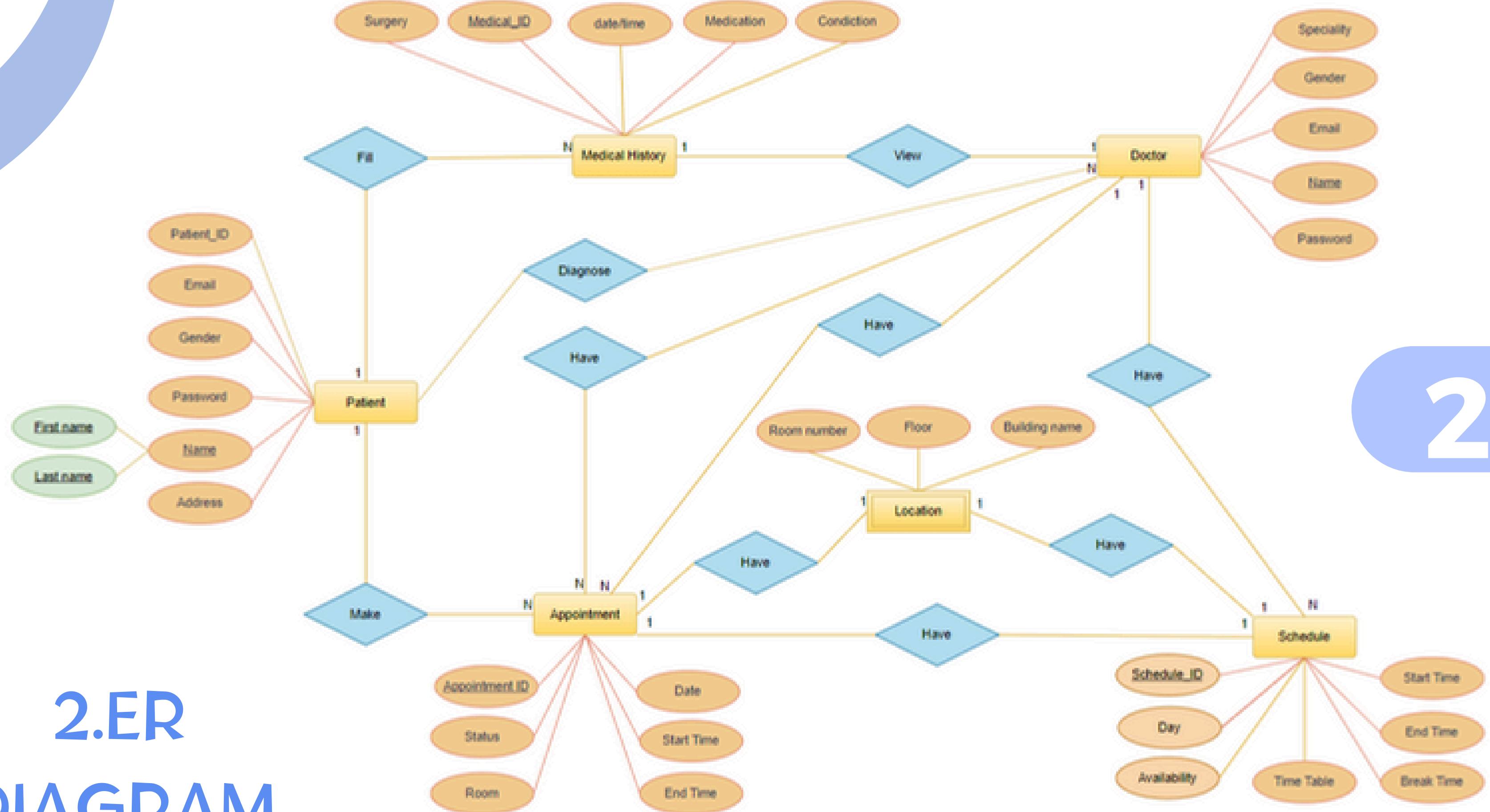
Conclusion

1.TEACHER RECOMMANDATION

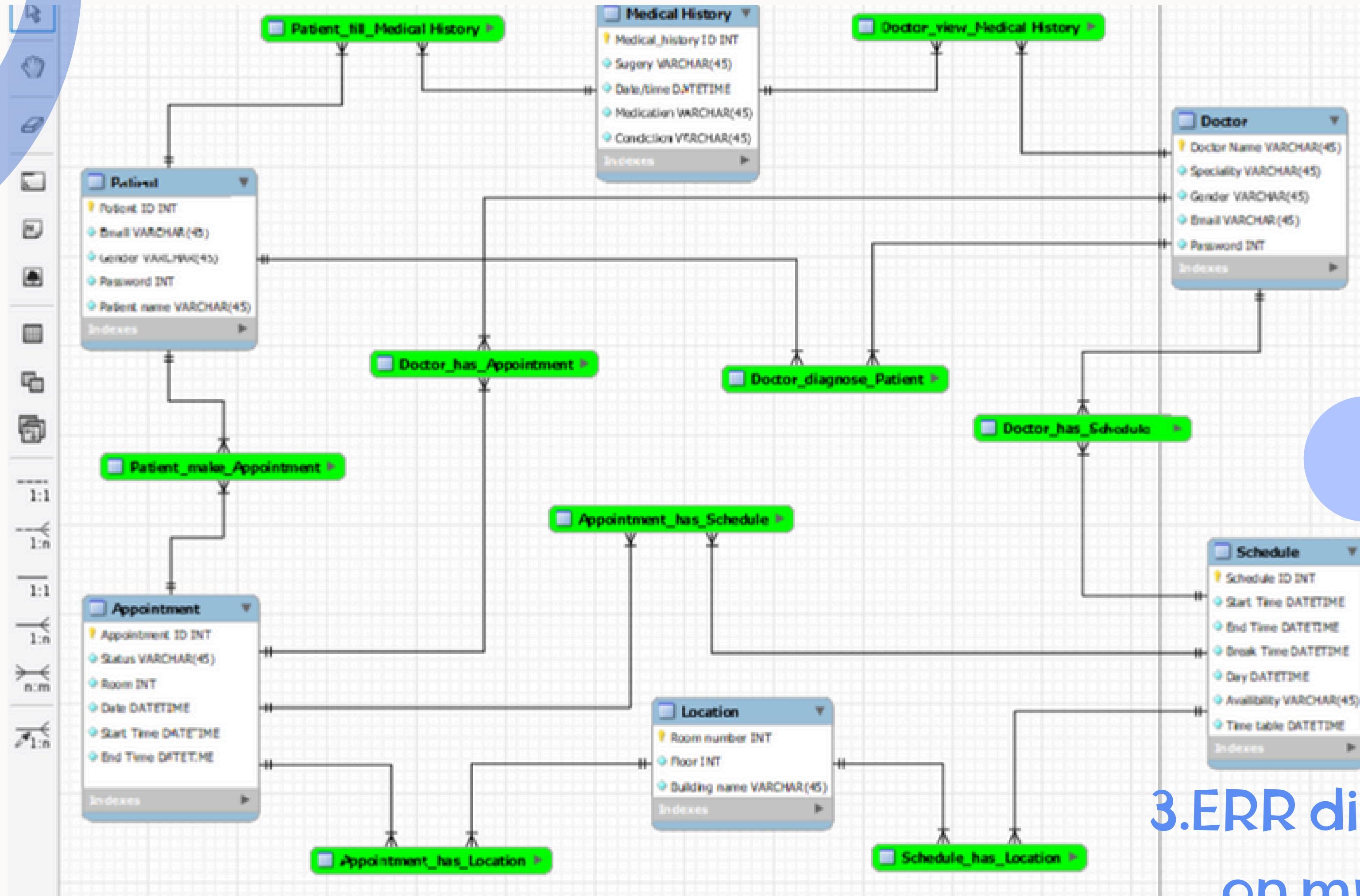
After last time presentation here are some things we need to add up on:

- 1. Patient_id(attributes) to patient (entity)**
- 2. Madical_id(attribute) to Medical history (entity)**
- 3. Additional attributes first name, last name to name(attribute) in Patient(entity)- “optional”**

2.ER DIAGRAM



3. ER diagram on mysql



CONCLUSION

In conclusion, employing an Entity-Relationship Diagram (ERD) on MySQL for a Hospital Management System (HMS) streamlines healthcare operations. The ERD organizes patient records, staff management, inventory control, and billing systems logically, enhancing decision-making and operational efficiency.



A cartoon illustration of a doctor with short dark hair and glasses, wearing a white coat and a stethoscope. The doctor is smiling and waving with one hand. Behind the doctor is a large blue circular graphic containing five blue speech bubbles with white icons: a clipboard, a medical cross, a capsule, an ECG line, and a stethoscope. The background features stylized blue and white leaf-like patterns.

**THANK YOU
VERY MUCH !**



WELCOME BACK

Group: 2 AMS-A

THIS WEEK CONTENTS

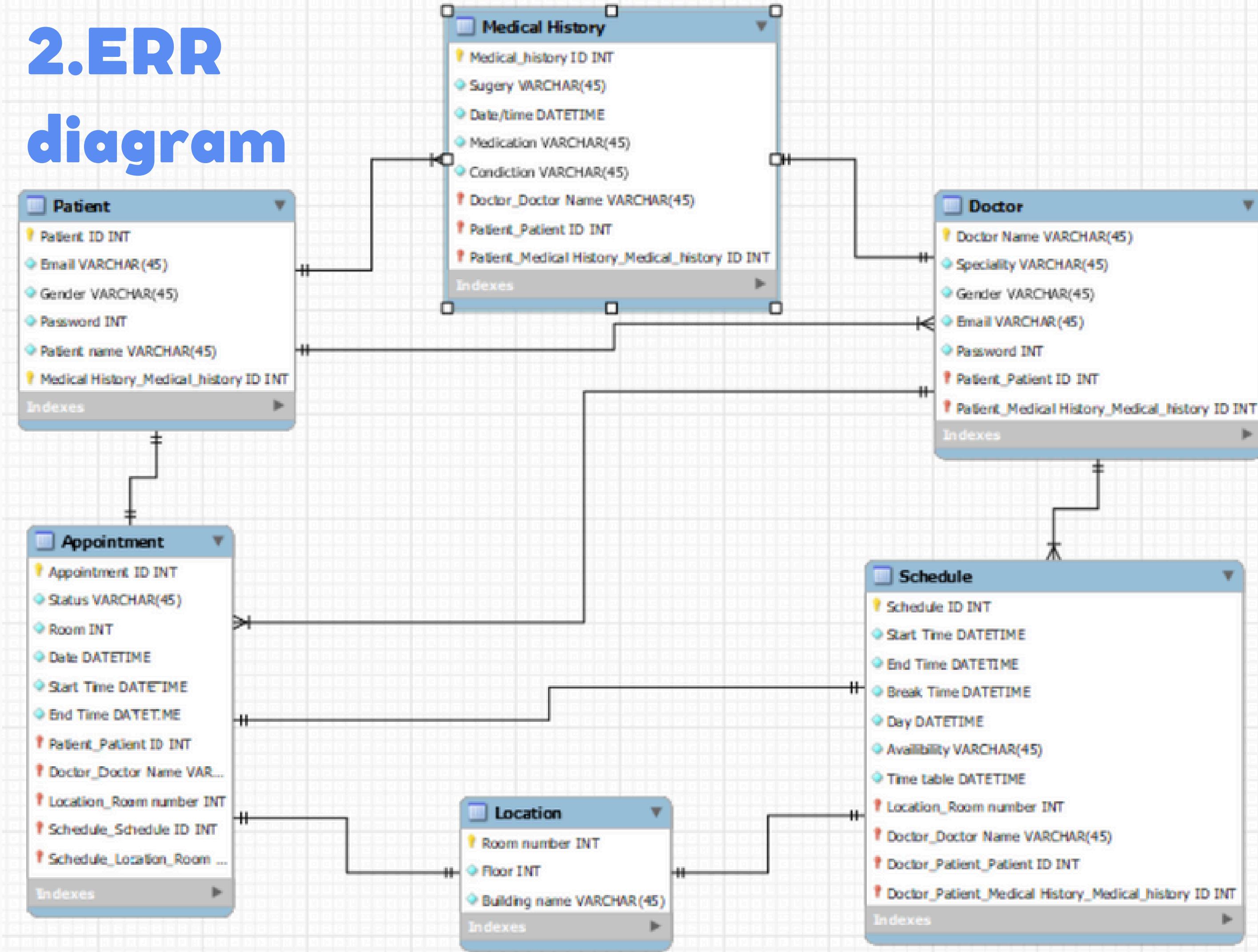
1

About Our
Improvement

2

Next Step

2.ERR diagram



2.PLAN FOR NEXT WEEK

- Code Connect Database
- SQL Code
- Update about Functional Dependency

A cartoon illustration of a doctor with dark hair and glasses, wearing a white coat and a stethoscope. The doctor is smiling and waving with one hand. Behind the doctor are four blue speech bubbles containing medical icons: a clipboard, a cross, a capsule, and an ECG line. The background is white with blue diagonal stripes.

**THANK YOU
VERY MUCH !**



WELCOME BACK

Group: 2 AMS-A

THIS WEEK CONTENTS

1

Last Week
Missing

2

Finalize ERR
diagram

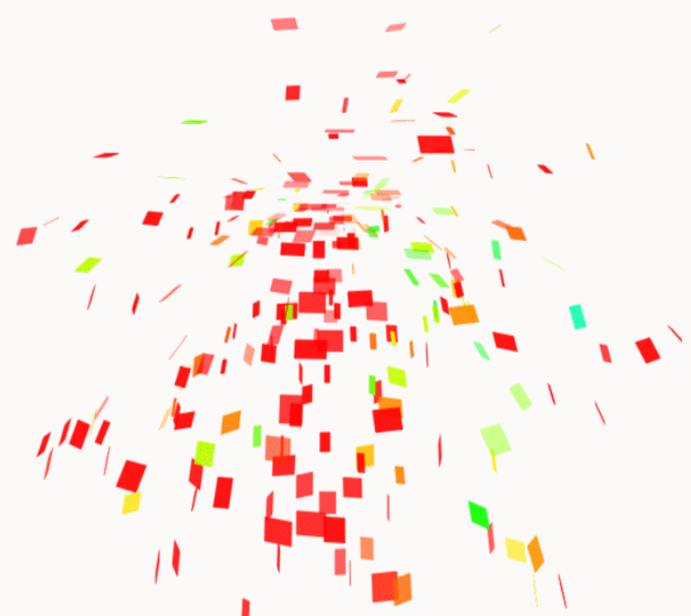
3

Functional
Dependency

33

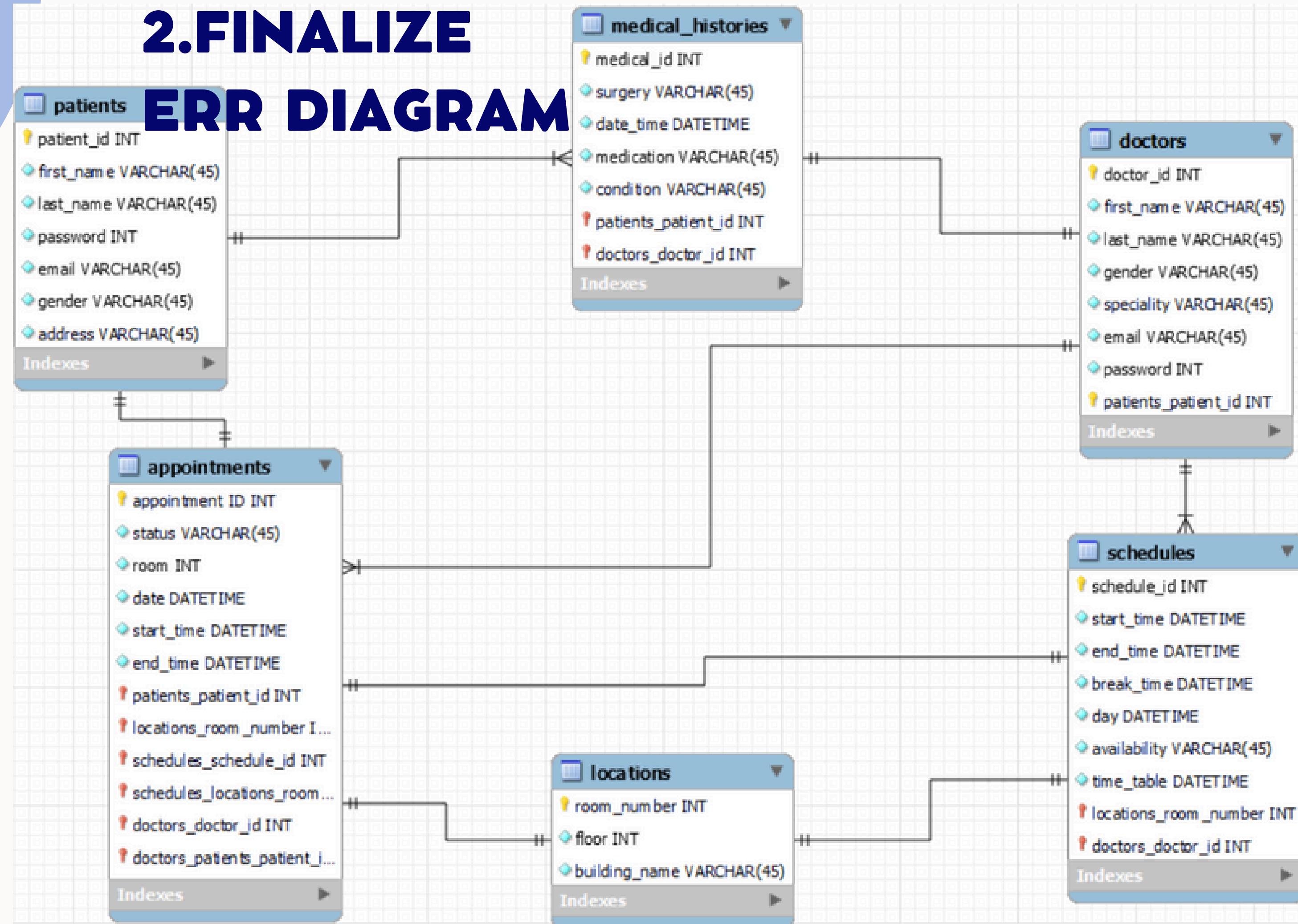
1. LAST WEEK MISSING

During last week's DBMS presentation, my team forgot to showcase sample data to our lecturer. We're rectifying this oversight by presenting the missing sample data this week. Additionally, we'll provide updates on functional dependency concepts to ensure a comprehensive understanding of the topic.



34

2.FINALIZE ERD DIAGRAM



35

SAMPLE DATA

Patient Table						
FirstName	LastName	Gender	Address	Email	Password	
Emily	Smith	Female	123 Maple Street, Springfield, IL	emilysmith@example.com	P@ssw0rd123	
John	Doe	Male	456 Oak Avenue, Anytown, NY	johndoe@example.com	Password123!	
Sarah	Johnson	Female	789 Pine Lane, Smalltown, CA	sarahjohnson@example.com	SecurePass456	
Michael	Williams	Male	101 Elm Street, Cityville, TX	michaelwilliams@example.com	Welcome@123	
Jennifer	Brown	Female	234 Cedar Road, Villageton, FL	jenniferbrown@example.com	SafePassword789	
David	Jones	Male	567 Birch Court, Countryside, OH	davidjones@example.com	P@55w0rd!	
Jessica	Davis	Female	890 Walnut Drive, Hamlet, WA	jessicadavis@example.com	Secret@Pass	
Christopher	Miller	Male	1234 Oakwood Lane, Suburbia, GA	christophermiller@example.com	P@sswordSecure	
Amanda	Martin	Female	5678 Maplewood Avenue, Townsville, MA	amandamartin@example.com	P@55w0rd!	
Matthew	Thompson	Male	91011 Pinecrest Road, Ruralville, NV	matthewthompson@example.com	Th0rn	
Superkey	{Name}, {Email}, {Name, Email}					
Candidate Key	{Email} (since it uniquely identifies each patient)					
Primary Key	{Email} (chosen from the candidate keys)					
Foreign Key	None					

SAMPLE DATA

Doctor					
Name	DoctorID	Gender	Speciality	Email	Password
Dr. Emily Smith	DOC123	Female	Cardiology	emilysmith@example.com	P@ssw0rd123
Dr. John Doe	DOC456	Male	Neurology	johndoe@example.com	Password123!
Dr. Sarah Johnson	DOC789	Female	Pediatrics	sarahjohnson@example.com	SecurePass456
Dr. Michael Williams	DOC101	Male	Orthopedics	michaelwilliams@example.com	Welcome@123
Dr. Jennifer Brown	DOC234	Female	Dermatology	jenniferbrown@example.com	SafePassword789
Dr. David Jones	DOC567	Male	Psychiatry	davidjones@example.com	P@55w0rd!
Dr. Jessica Davis	DOC890	Female	Oncology	jessicadavis@example.com	Secret@Pass
Dr. Christopher Miller	DOC111	Male	Cardiac Surgery	christophermiller@example.com	P@sswordSecure
Dr. Amanda Martin	DOC222	Female	Gynecology	amandamartin@example.com	P@55w0rd!
Dr. Matthew Thompson	DOC333	Male	Urology	matthewthompson@example.com	Th0rnsR0s3!

Superkey	{Name}, {DoctorID}, {Email}, {Name, Email}
Candidate Key	{Email}, {DoctorID} (since it uniquely identifies each doctor)
Primary Key	{Email}, {DoctorID} (chosen from the candidate keys)
Foreign Key	None

SAMPLE DATA

Medical History				
Medical ID	Date/Time	Surgery	Medication	Condition
MEDID001	5/5/2024 8:00	Appendectomy	Ibuprofen 400mg	Appendicitis
MEDID002	5/6/2024 10:30	Knee Replacement	Oxycodone 5mg	Osteoarthritis
MEDID003	5/7/2024 14:15	Cataract Surgery	Prednisolone Eye Drops	Cataracts
MEDID004	5/8/2024 9:45	Gallbladder Removal	Ursodiol 300mg	Cholecystitis
MEDID005	5/9/2024 11:20	Hip Arthroscopy	Morphine 10mg	Hip Impingement
MEDID006	5/10/2024 13:40	Tonsillectomy	Acetaminophen 500mg	Tonsillitis
MEDID007	5/11/2024 15:00	Coronary Angioplasty	Aspirin 81mg	Coronary Artery Disease
MEDID008	5/12/2024 7:55	Lumbar Fusion Surgery	Gabapentin 300mg	Degenerative Disc Disease
MEDID009	5/13/2024 10:10	Hysterectomy	Estrogen Replacement Therapy	Uterine Fibroids
MEDID010	5/14/2024 12:30	Rhinoplasty	Antibiotic Ointment	Deviated Septum

Superkey	{Medical ID}		
Candidate Key	{Medical ID} (since it uniquely identifies each medical history entry)		
Primary Key	{Medical ID} (chosen from the candidate keys)		
Foreign Key	None		

SAMPLE DATA

Appointment						
Appointment ID	Date	Start Time	End Time	Status	Room	
APPTID001	5/5/2024	9:00	10:00	Scheduled	Room 1	
APPTID002	5/6/2024	10:30	11:30	Confirmed	Room 2	
APPTID003	5/7/2024	13:00	14:00	Scheduled	Room 3	
APPTID004	5/8/2024	11:30	12:30	Confirmed	Room 4	
APPTID005	5/9/2024	14:00	15:00	Scheduled	Room 5	
APPTID006	5/10/2024	15:30	16:30	Confirmed	Room 6	
APPTID007	5/11/2024	8:30	9:30	Scheduled	Room 7	
APPTID008	5/12/2024	9:45	10:45	Confirmed	Room 8	
APPTID009	5/13/2024	12:00	13:00	Scheduled	Room 9	
APPTID010	5/14/2024	13:30	14:30	Confirmed	Room 10	

Superkey	{Appointment ID}, {Date, Start Time}, {Date, Room}, {Date, Room, Start Time}, {Date, Room, Start Time, End Time}
Candidate Key	{Appointment ID}, {Date, Start Time} (both uniquely identify each appointment)
Primary Key	{Appointment ID} (chosen from the candidate keys)
Foreign Key	{Room} (references Location(RoomNumber))

SAMPLE DATA

Location		
RoomNumber	Floor	Building Name
Room 101	1	A
Room 202	2	B
Room 303	3	C
Room 404	4	D
Room 505	5	E
Room 606	6	F
Room 707	7	G
Room 808	8	H
Room 909	9	I
Room 1010	10	J

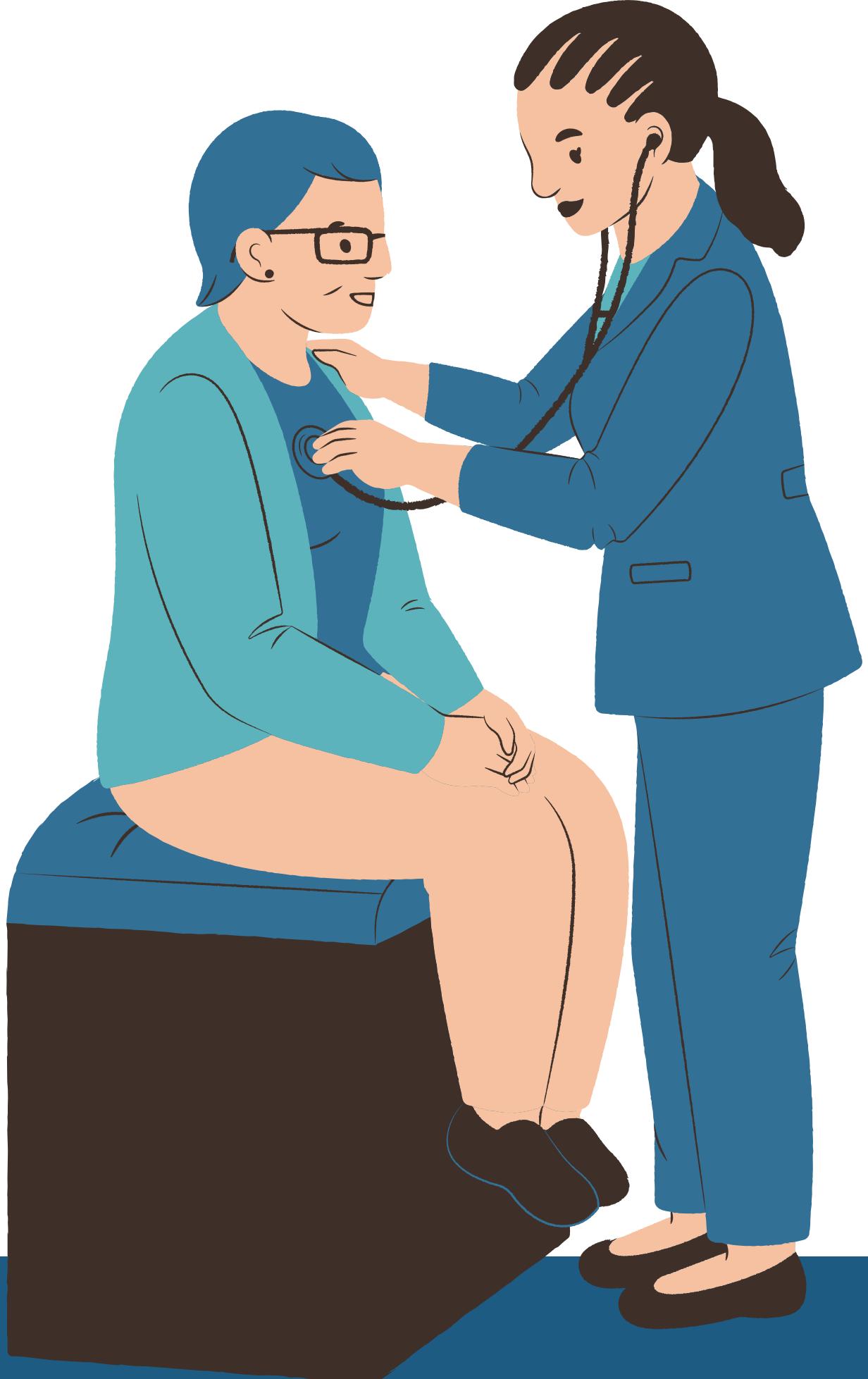
Superkey	{RoomNumber}, {Floor, RoomNumber}
Candidate Key	{RoomNumber} (since it uniquely identifies each location)
Primary Key	{RoomNumber} (chosen from the candidate keys)
Foreign Key	None

3. FUNCTIONAL DEPENDENCY

Patient Table:	Fully Dependent Attribute: $Email \rightarrow FirstName, LastName, Gender, Address, Password$ Partially Dependent Attribute: Insurance Provider (dependent on Address): $Address \rightarrow Insurance\ Provider$
Doctor Table:	Fully Dependent Attribute: $DoctorID \rightarrow Name, Gender, Speciality, Email, Password$ Partially Dependent Attribute: Years of Experience (dependent on Age): $Age \rightarrow Years\ of\ Experience$
Medical History Table:	Fully Dependent Attribute: $Medical\ ID \rightarrow Date/Time, Surgery, Medication, Condition$ No Partially Dependent Attribute
Appointment Table:	Fully Dependent Attribute: $Appointment\ ID \rightarrow Date, Start\ Time, End\ Time, Status, Room$ Partially Dependent Attribute: Appointment Type (dependent on Doctor Speciality): $Doctor\ Speciality \rightarrow Appointment\ Type$
Location Table:	Fully Dependent Attribute: $RoomNumber \rightarrow Floor, Building\ Name$ No Partially Dependent Attribute

CONCLUSION

In conclusion, we have rectified the omission of sample data from last week's presentation and provided a comprehensive update on functional dependency concepts. Our efforts aim to enhance understanding and deliver a more thorough presentation this week.



Thank you!



AMS, ITC, Phnom Penh



Welcome back Once again

Group: 2 AMS-A

FOR TODAY CONTENTS

1

Normalization

2

Conclusion

42

1.NORMALIZATION

Normalization is a process by which data structure in a relational database are as efficient as possible, including

- The elimination of redundancy
- elimination of redundancy
- The prevention of the loss of information

Base on our sample dataset that we did last week. we will normalize on entity patient, doctor and appointment as those main entity have more attributes.

I.PATIENTS

patients							
patient_id	first_name	last_name	gender	address	email	password	
PAT00001	Emily	Smith	Female	123 Maple Street, Springfield, IL	emilysmith@example.com	P@ssw0rd123	
PAT00002	John	Doe	Male	456 Oak Avenue, Anytown, NY	johndoe@example.com	Password123!	
PAT00003	Sarah	Johnson	Female	789 Pine Lane, Smalltown, CA	sarahjohnson@example.com	SecurePass456	
PAT00004	Michael	Williams	Male	101 Elm Street, Cityville, TX	michaelwilliams@example.com	Welcome@123	
PAT00005	Jennifer	Brown	Female	234 Cedar Road, Villageton, FL	jenniferbrown@example.com	SafePassword789	
PAT00006	David	Jones	Male	567 Birch Court, Countryside, OH	davidjones@example.com	P@55w0rd!	
PAT00007	Jessica	Davis	Female	890 Walnut Drive, Hamlet, WA	jessicadavis@example.com	Secret@Pass	
PAT00008	Christopher	Miller	Male	1234 Oakwood Lane, Suburbia, GA	christophermiller@example.com	P@sswordSecure	
PAT00009	Amanda	Martin	Female	5678 Maplewood Avenue, Townsville, MA	amandamartin@example.com	P@55w0rd!	
PAT00010	Matthew	Thompson	Male	91011 Pinecrest Road, Ruralville, NV	matthewthompson@example.com	Th0rn	

- Normalize patients(entity) :
- patient_Info(patient_id,first_name,last_name)
- patient_mail(email,password,address)

patients(patient_info,patient_mail,gender)

II. DOCTORS

doctors						
first_name	last_name	doctor_id	gender	speciality	email	password
Dr. Emily	Smith	DOC123	Female	Cardiology	emilysmith@example.com	P@ssw0rd123
Dr. John	Doe	DOC456	Male	Neurology	johndoe@example.com	Password123!
Dr. Sarah	Johnson	DOC789	Female	Pediatrics	sarahjohnson@example.com	SecurePass456
Dr. Michael	Williams	DOC101	Male	Orthopedics	michaelwilliams@example.com	Welcome@123
Dr. Jennifer	Brown	DOC234	Female	Dermatology	jenniferbrown@example.com	SafePassword789
Dr. David	Jones	DOC567	Male	Psychiatry	davidjones@example.com	P@55w0rd!
Dr. Jessica	Davis	DOC890	Female	Oncology	jessicadavis@example.com	Secret@Pass
Dr. Christopher	Miller	DOC111	Male	Cardiac Surgery	christophermiller@example.com	P@sswordSecure
Dr. Amanda	Martin	DOC222	Female	Gynecology	amandamartin@example.com	P@55w0rd!
Dr. Matthew	Thompson	DOC333	Male	Urology	matthewthompson@example.com	Th0rnsR0s3!

- Normalize doctors(entity) :
- doctor_Info(first_name, last_name, doctor_id, speciality)
 doctor_mail(email, password)

doctors(doctor_info,doctor_mail,gender)

III. APPOINTMENTS

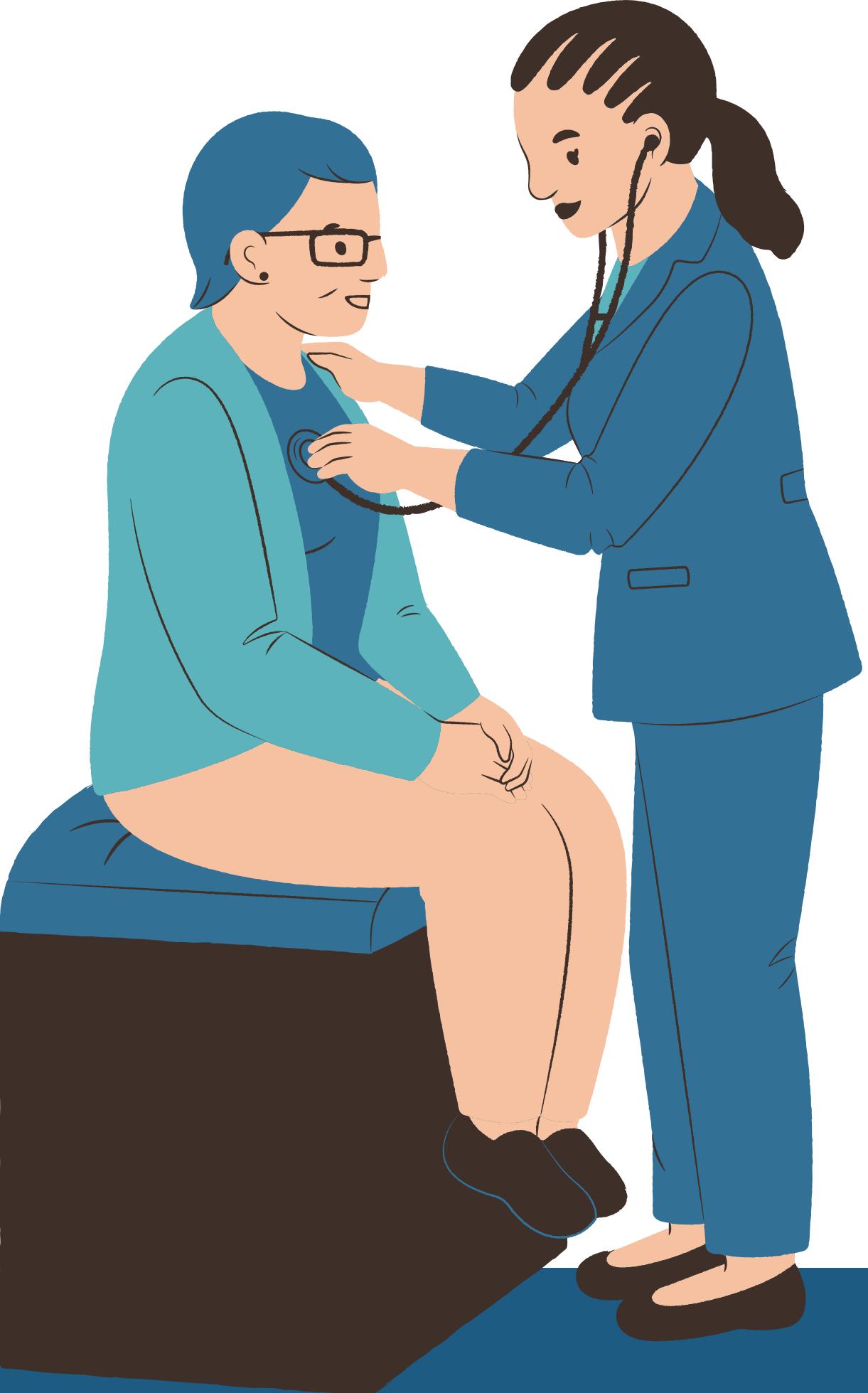
appointments	appointment_id	date	start_time	end_time	status	room
	APPTID001	5/5/2024	9:00	10:00	Scheduled	Room 1
	APPTID002	5/6/2024	10:30	11:30	Confirmed	Room 2
	APPTID003	5/7/2024	13:00	14:00	Scheduled	Room 3
	APPTID004	5/8/2024	11:30	12:30	Confirmed	Room 4
	APPTID005	5/9/2024	14:00	15:00	Scheduled	Room 5
	APPTID006	5/10/2024	15:30	16:30	Confirmed	Room 6
	APPTID007	5/11/2024	8:30	9:30	Scheduled	Room 7
	APPTID008	5/12/2024	9:45	10:45	Confirmed	Room 8
	APPTID009	5/13/2024	12:00	13:00	Scheduled	Room 9
	APPTID010	5/14/2024	13:30	14:30	Confirmed	Room 10

- Normalize appointments(entity) :
appointment_detail(appointment_id,date)
appointment_times(start_time,end_time)

appointments(appointment_date,appointment_times,status,room)

CONCLUSION

Overall, normalizing sample data in a hospital management system improves database efficiency, reduces data redundancy, and enhances data integrity, ultimately contributing to the effective management of hospital operations and patient care.



Thank you!



AMS, ITC, Phnom Penh



WELCOME BACK AGAIN

Group: 2 AMS-A

THIS WEEK CONTENTS

1

Overview on
ER diagram

2

SQL Query 1

3

Insert Query

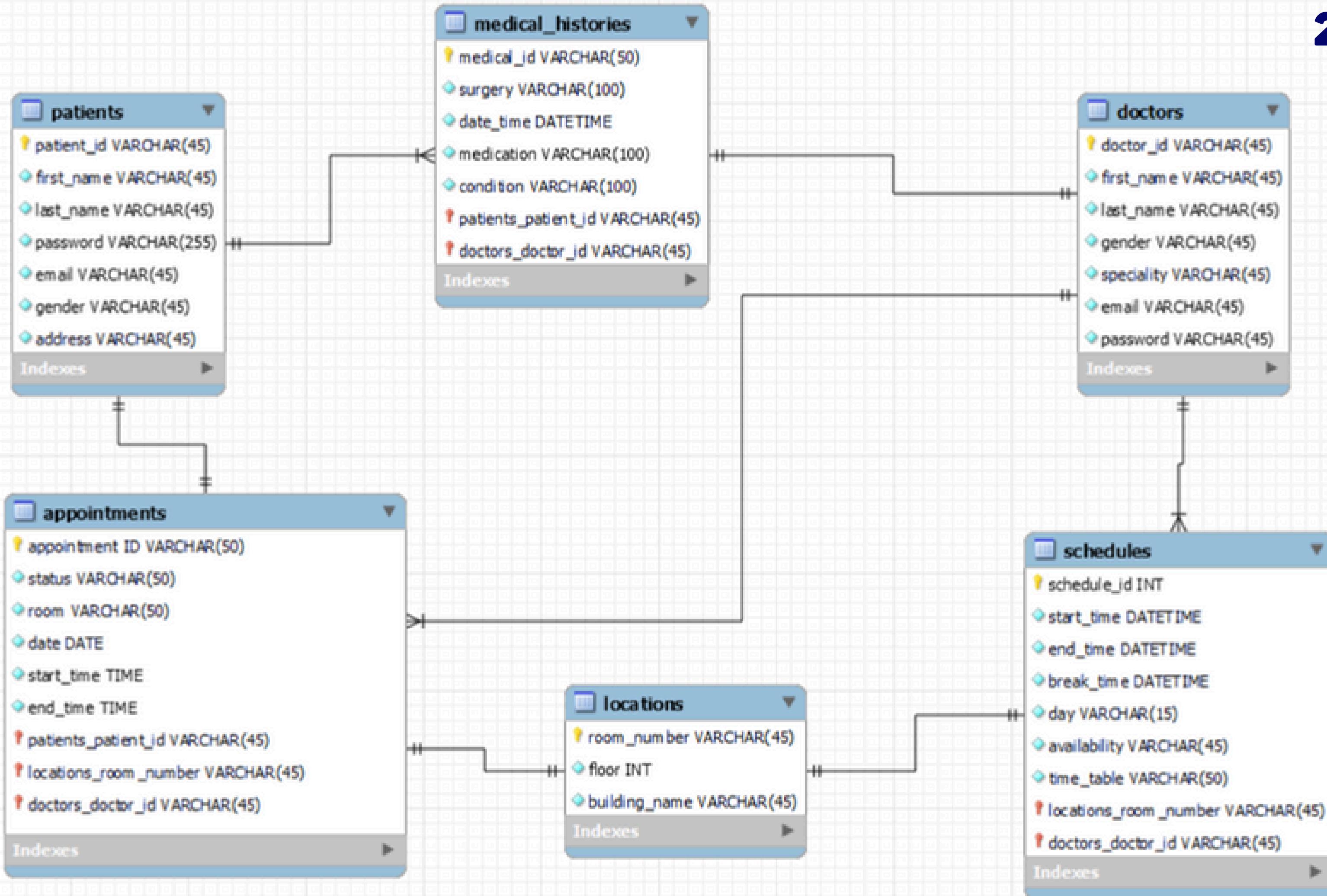
4

Select Query

47

2.FINALIZE ERR DIAGRAM

48



2.SQL QUERY 1

Query 1 ×

The screenshot shows a MySQL Workbench interface with a query editor titled "Query 1". The code is as follows:

```
1 • DROP DATABASE IF EXISTS data;
2
3 • CREATE DATABASE data;
4 • USE data;
5
6 • CREATE TABLE Patients(
    patient_id VARCHAR(45), -- Correct data type for alphanumeric IDs
    first_name VARCHAR(45),
    last_name VARCHAR(45),
    password VARCHAR(255), -- Changed to VARCHAR to accommodate hashed or
                           alphanumeric passwords
    email VARCHAR(45),
    gender VARCHAR(45),
    address VARCHAR(45),
    PRIMARY KEY (patient_id)
)j
```

The code includes comments explaining changes made to the schema. A vertical line with a circle at the top points from the line number 6 to the opening parenthesis of the table definition.

```
19 • Ⓛ CREATE TABLE Doctors(  
20     doctor_id VARCHAR(45),  
21     first_name VARCHAR(45),  
22     last_name VARCHAR(45),  
23     gender VARCHAR(45),  
24     speciality VARCHAR(45),  
25     email VARCHAR(45),  
26     password VARCHAR(45),  
27     primary key (doctor_id)  
28 );  
29  
30 • Ⓛ CREATE TABLE Location(  
31     room_number VARCHAR(45),  
32     floor INT,  
33     building_name VARCHAR(45),  
34     PRIMARY KEY (room_number)  
35 );  
36
```

```
• CREATE TABLE medical_histories (
    medical_id VARCHAR(50) PRIMARY KEY,
    date_time DATETIME,
    surgery VARCHAR(100),
    medication VARCHAR(100),
    `condition` VARCHAR(100),
    patient_id VARCHAR(45),
    doctor_id VARCHAR(45),
    FOREIGN KEY (patient_id) REFERENCES Patients(patient_id),
    FOREIGN KEY (doctor_id) REFERENCES Doctors(doctor_id)
);
```

```
50 • CREATE TABLE schedules (
    schedule_id INT PRIMARY KEY,
    start_time TIME,
    end_time TIME,
    break_time VARCHAR(50), -- Assuming break_time is a range and not a specific
    DATETIME
    day VARCHAR(15), -- Assuming 'day' should be a day of the week
    availability VARCHAR(45),
    time_table VARCHAR(50), -- Assuming this is a range or description
    locations_room_number VARCHAR(45), -- Changed to match Location's room_number type
    doctors_doctor_id VARCHAR(45), -- Ensuring doctor_id type matches the
    Doctors table
    FOREIGN KEY (locations_room_number) REFERENCES Location(room_number),
    FOREIGN KEY (doctors_doctor_id) REFERENCES Doctors(doctor_id)
);
```

```
65 • CREATE TABLE appointments (
66     appointment_id VARCHAR(50) PRIMARY KEY,
67     date DATE,
68     start_time TIME,
69     end_time TIME,
70     status VARCHAR(50),
71     room VARCHAR(50),
72     patient_id VARCHAR(50),
73     doctor_id VARCHAR(50),
74     schedule_id INT,
75     FOREIGN KEY (patient_id) REFERENCES Patients(patient_id),
76     FOREIGN KEY (doctor_id) REFERENCES Doctors(doctor_id),
77     FOREIGN KEY (schedule_id) REFERENCES schedules(schedule_id)
78 );
```

3. INSERT QUERY

```
File Edit View Insert Tools Help | Limit to 1000 rows |            
```

```
1 • INSERT INTO patients (patient_id, first_name, last_name, gender, address, email, password)
2 VALUES
3 ('PAT00001', 'Emily', 'Smith', 'Female', '123 Maple Street, Springfield, IL', 'emilysmith@example.com', '123'),
4 ('PAT00002', 'John', 'Doe', 'Male', '456 Oak Avenue, Anugton, NY', 'johndoe@example.com', 'Password123!'),
5 ('PAT00003', 'Sarah', 'Johnson', 'Female', '789 Pine Lane, Smalltown, CA', 'sarahjohnson@example.com', 'SecurePa$$456'),
6 ('PAT00004', 'Michael', 'Williams', 'Male', '101 Elm Street, Cityville, TX', 'michaelwilliams@example.com', 'Welcome@123'),
7 ('PAT00005', 'Jennifer', 'Brown', 'Female', '234 Cedar Road, Villageton, FL', 'jenniferbrown@example.com', 'SafePassword789'),
8 ('PAT00006', 'David', 'Jones', 'Male', '567 Birch Court, Countryside, OH', 'davidjones@example.com', 'P@55w0rd!'),
9 ('PAT00007', 'Jessica', 'Davis', 'Female', '890 Walnut Drive, Hamlet, MA', 'jessicadavis@example.com', 'Secret@Pass'),
10 ('PAT00008', 'Christopher', 'Miller', 'Male', '1234 Oakwood Lane, Suburbia, GA', 'christophermiller@example.com', 'PasswordSecure'),
11 ('PAT00009', 'Amanda', 'Martin', 'Female', '5678 Maplewood Avenue, Townsville, MA', 'amandamartin@example.com', 'P@55w0rd!'),
12 ('PAT00010', 'Matthew', 'Thompson', 'Male', '91011 Pinecrest Road, Ruralville, NY', 'matthewthompson@example.com', 'Th0rn')
--
```

```
16 • INSERT INTO Doctors (doctor_id, first_name, last_name, gender, speciality, email, password)
17 VALUES
18 ('DOC123', 'Dr. Emily', 'Smith', 'Female', 'Cardiology', 'emilysmith@example.com', 'P@ssw0rd123'),
19 ('DOC456', 'Dr. John', 'Doe', 'Male', 'Neurology', 'johndoe@example.com', 'Password123!'),
20 ('DOC789', 'Dr. Sarah', 'Johnson', 'Female', 'Pediatrics', 'sarahjohnson@example.com', 'SecurePass456'),
21 ('DOC101', 'Dr. Michael', 'Williams', 'Male', 'Orthopedics', 'michaelwilliams@example.com', 'Welcome@123'),
22 ('DOC234', 'Dr. Jennifer', 'Brown', 'Female', 'Dermatology', 'jenniferbrown@example.com', 'SafePassword789'),
23 ('DOC567', 'Dr. David', 'Jones', 'Male', 'Psychiatry', 'davidjones@example.com', 'P@55w0rd!'),
24 ('DOC890', 'Dr. Jessica', 'Davis', 'Female', 'Oncology', 'jessicadavis@example.com', 'Secret@Pass'),
25 ('DOC111', 'Dr. Christopher', 'Miller', 'Male', 'Cardiac Surgery', 'christophermiller@example.com', 'P@sswordSecure'),
26 ('DOC222', 'Dr. Amanda', 'Martin', 'Female', 'Gynecology', 'amandamartin@example.com', 'P@55w0rd!'),
27 ('DOC333', 'Dr. Matthew', 'Thompson', 'Male', 'Urology', 'matthewthompson@example.com', 'Th0rn');
```

```
21  
22  
23  
24  
25  
26  
27  
28  
29 • INSERT INTO Location (room_number, floor, building_name)  
30     VALUES  
31     ('Room 101', 1, 'A'),  
32     ('Room 202', 2, 'B'),  
33     ('Room 303', 3, 'C'),  
34     ('Room 404', 4, 'D'),  
35     ('Room 505', 5, 'E'),  
36     ('Room 606', 6, 'F'),  
37     ('Room 707', 7, 'G'),  
38     ('Room 808', 8, 'H'),  
39     ('Room 909', 9, 'I'),  
40     ('Room 1010', 10, 'J');  
  
42 • INSERT INTO medical_histories (medical_id, date_time, surgery, medication, `condition`, patient_id, doctor_id)  
43     VALUES  
44     ('MEDID001', '2024-05-05 08:00', 'Appendectomy', 'Ibuprofen 400mg', 'Appendicitis', 'PAT00001', 'DOC123'),  
45     ('MEDID002', '2024-05-06 10:30', 'Knee Replacement', 'Oxycodone 5mg', 'Osteoarthritis', 'PAT00002', 'DOC456'),  
46     ('MEDID003', '2024-05-07 14:15', 'Cataract Surgery', 'Prednisolone Eye Drops', 'Cataracts', 'PAT00003', 'DOC789'),  
47     ('MEDID004', '2024-05-08 09:45', 'Gallbladder Removal', 'Ursodiol 300mg', 'Cholecystitis', 'PAT00004', 'DOC101'),  
48     ('MEDID005', '2024-05-09 11:20', 'Hip Arthroscopy', 'Morphine 10mg', 'Hip Impingement', 'PAT00005', 'DOC234'),  
49     ('MEDID006', '2024-05-10 13:40', 'Tonsillectomy', 'Acetaminophen 500mg', 'Tonsillitis', 'PAT00006', 'DOC567'),  
50     ('MEDID007', '2024-05-11 15:00', 'Coronary Angioplasty', 'Aspirin 81mg', 'Coronary Artery Disease', 'PAT00007', 'DOC890'),  
51     ('MEDID008', '2024-05-12 07:55', 'Lumbar Fusion Surgery', 'Gabapentin 300mg', 'Degenerative Disc Disease', 'PAT00008', 'DOC111'),  
52     ('MEDID009', '2024-05-13 10:10', 'Hysterectomy', 'Estrogen Replacement Therapy', 'Uterine Fibroids', 'PAT00009', 'DOC222'),  
53     ('MEDID010', '2024-05-14 12:30', 'Rhinoplasty', 'Antibiotic Ointment', 'Deviated Septum', 'PAT00010', 'DOC333');
```

```
55 • INSERT INTO schedules (schedule_id, start_time, end_time, break_time, day, availability, time_table, doctors_doctor_id)
56 VALUES
57 (1001, '08:00', '16:00', '12:00 - 12:30', 'Monday', 'Yes', '08:00 - 16:00', 'DOC123'),
58 (1002, '09:00', '17:00', '13:00 - 13:30', 'Tuesday', 'No', '09:00 - 17:00', 'DOC456'),
59 (1003, '10:00', '18:00', '14:00 - 14:30', 'Wednesday', 'Yes', '10:00 - 18:00', 'DOC789'),
60 (1004, '11:00', '19:00', '15:00 - 15:30', 'Thursday', 'No', '11:00 - 19:00', 'DOC101'),
61 (1005, '12:00', '20:00', '16:00 - 16:30', 'Friday', 'Yes', '12:00 - 20:00', 'DOC234'),
62 (1006, '13:00', '21:00', '17:00 - 17:30', 'Saturday', 'Yes', '13:00 - 21:00', 'DOC567'),
63 (1007, '14:00', '22:00', '18:00 - 18:30', 'Sunday', 'No', '14:00 - 22:00', 'DOC890'),
64 (1008, '15:00', '23:00', '19:00 - 19:30', 'Monday', 'Yes', '15:00 - 23:00', 'DOC111'),
65 (1009, '16:00', '00:00', '20:00 - 20:30', 'Tuesday', 'No', '16:00 - 00:00', 'DOC222'),
66 (1010, '17:00', '01:00', '21:00 - 21:30', 'Wednesday', 'Yes', '17:00 - 01:00', 'DOC333');
```

```
68 • INSERT INTO appointments (appointment_id, date, start_time, end_time, status, room, patient_id, doctor_id, schedule_id)
69 VALUES
70 ('APPTID001', '2024-05-05', '09:00', '10:00', 'Scheduled', 'Room 1', 'PAT00001', 'DOC123', 1001),
71 ('APPTID002', '2024-05-06', '10:30', '11:30', 'Confirmed', 'Room 2', 'PAT00002', 'DOC456', 1002),
72 ('APPTID003', '2024-05-07', '13:00', '14:00', 'Scheduled', 'Room 3', 'PAT00003', 'DOC789', 1003),
73 ('APPTID004', '2024-05-08', '11:30', '12:30', 'Confirmed', 'Room 4', 'PAT00004', 'DOC101', 1004),
74 ('APPTID005', '2024-05-09', '14:00', '15:00', 'Scheduled', 'Room 5', 'PAT00005', 'DOC234', 1005),
75 ('APPTID006', '2024-05-10', '15:30', '16:30', 'Confirmed', 'Room 6', 'PAT00006', 'DOC567', 1006),
76 ('APPTID007', '2024-05-11', '08:30', '09:30', 'Scheduled', 'Room 7', 'PAT00007', 'DOC890', 1007),
77 ('APPTID008', '2024-05-12', '09:45', '10:45', 'Confirmed', 'Room 8', 'PAT00008', 'DOC111', 1008),
78 ('APPTID009', '2024-05-13', '12:00', '13:00', 'Scheduled', 'Room 9', 'PAT00009', 'DOC222', 1009),
79 ('APPTID010', '2024-05-14', '13:30', '14:30', 'Confirmed', 'Room 10', 'PAT00010', 'DOC333', 1010);
```

4.SELECT QUERY

```
1 • select
2     patient_id,
3     first_name,
4     last_name
5 from
6     patients
7 order by
8     first_name DESC,
9     last_name ASC
10 limit 5
```

The screenshot shows a MySQL Workbench result grid. At the top, there are buttons for 'Result Grid' (selected), 'Filter Rows:', 'Export', 'Wrap Cell Content:', and 'Fetch rows:'. The result grid displays a table with columns: patient_id, first_name, and last_name. The data rows are:

	patient_id	first_name	last_name
▶	PAT00003	Sarah	Johnson
	PAT00004	Michael	Williams
	PAT00010	Matthew	Thompson
	PAT00002	John	Doe
	PAT00007	Jessica	Davis

```
1 •   select
2       patient_id,
3       first_name,
4       last_name
5   from
6       patients
7   where
8       first_name like '%A%'
```

Result Grid | Filter Rows: _____ | Export: Wrap Cell Content:

	patient_id	first_name	last_name
▶	PAT00003	Sarah	Johnson
	PAT00004	Michael	Williams
	PAT00006	David	Jones
	PAT00007	Jessica	Davis
	PAT00009	Amanda	Martin
	PAT00010	Matthew	Thompson

```
1 • select  
2     patient_id,  
3     first_name,  
4     last_name  
5 from  
6     patients  
7 where  
8     first_name like 'A%'
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

	patient_id	first_name	last_name
▶	PAT00009	Amanda	Martin

**THANK YOU
VERY MUCH !**

