



# Hospital Management System

By: Ligia | Meenal | Rawaf



# TOC

- ☐ Introduction
- ☐ Problem Description and Solution
- ☐ Defining Entities and Attributes
  - ☐ Entity Relationship Diagram
- ☐ Relational Database
  - ☐ Relational Design
  - ☐ Constraints
  - ☐ Validation Rules
- ☐ Queries
- ☐ Report



# Introduction

River Edge Hospital located in New Jersey, Bergen county. They offer a comprehensive range of services to patients in a caring, culturally sensitive environment, accessing all of the specialties and resources. It's the only care facility in the town of River Edge that has 300 staff members with over 100 patients are admitted and registered every day. As a small local hospital that has limited staff members and large incoming patients daily maintain their database system is mandatory.





# Problem Description

1

Operating on the old paper system is not very efficient or reliable. Has a lot of setbacks and problems to medical staff & patients throughout the years.

2

The process of the old system is very time consuming that will not be helpful in an environment where time is extremely valuable.

3

Old sys is not a user-friendly or easy to use system. Records of inpatient and outpatient are not easily tracked and get overlapped.

4

Large number of patients, doctors & staff members make data processing more crucial to be organized in a system that will make it easy to access.



## Project Objective & Solution

- ☐ The main goal of automating hospital systems is to minimize the paperwork as minimum as possible.
- ☐ Hospitals need a system that will enable their management to make effective and efficient decisions.
- ☐ Help to easily track patients data, type of treatment they got. Also helps doctors to track their treated patients data.
- ☐ Help in maintaining patient's information, arranging doctors & nurses' schedule, accessing patient's records.
- ☐ Also help patients to access their medical files, lab results and information about their case and diagnosis.



# Defining Entities and Attributes


An Entity Relationship Model (ERM) is the result of systematic analysis to define and describe what is important to process in an area of business. However, it does not define the business processes; it only presents a business data schema in graphical form. Also, ERM defines a data or information structure implemented in a database, typically a relational database. Through graphic visualization, Entity Relationship diagram shows the various entities and relations between entities.

## Classes and Attributes from Hospital Management System




### Patient

\*

 Patient\_ID  
Patient\_Name  
Patient\_Email  
Patient\_Phone  
Patient\_Address  
Patient\_DOB  
Patient\_Gender  
Patient\_Room\_No


### Medicine

\*

 Medicine\_ID  
medicine\_name  
Medicine\_Description  
Medicine\_Cost



### Treatment

\*

 Treatment\_ID  
Tretament\_Diagnosis



### Patient\_Record

\*

 Patient\_ID  
 Doctor\_ID  
Medicine\_ID  
Treatment\_ID


### Appointment

\*

 Patient\_ID  
 Doctor\_ID  
Appointment\_Date


### Doctor

\*

 Doctor\_ID  
Doctor\_Name  
Doctor\_Specialist  
Doctor\_Email  
Doctor\_Phone  
Doctor\_Work\_Day  
Doctor\_Work\_Start\_Time  
Doctor\_Work\_End\_Time  
Department\_ID

### Department

\*

 Department\_ID  
Department\_Name



# One to One Relationship

## 1 : 1

A one-to-one relationship is a type of cardinality that refers to the relationship between two entities A and B in which *one element of A may only be linked to one element of B*, and vice versa. For instance, think of A as the set of all countries, and B as the set of all their capital city. Any country from A can and must have only one capital city from B, and vice versa.

**In this database, no relationship is a one to one relationship.**





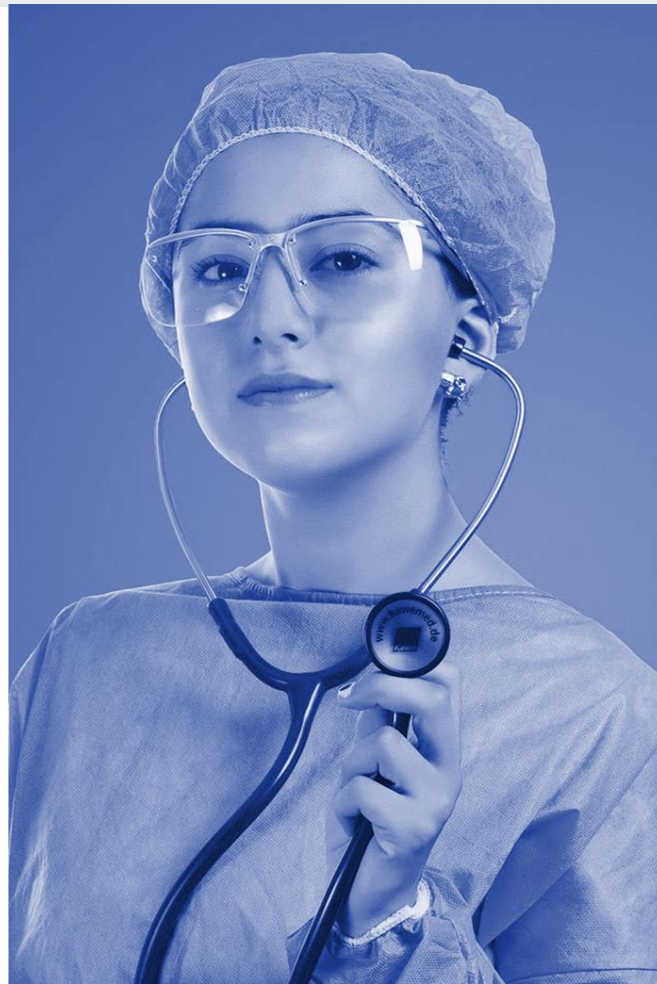


# One to Many Relationships

## 1 : N

A one-to-many relationship is a type of cardinality that refers to the relationship between two entities A and B in which *an element of A may be linked to many elements of B*, but a member of B is linked to only one element of A. For instance, think of A as mother, and B as kids. A mother can have many kids, but a kid can only have one mother.

- ❖ PATIENT 1:N APPOINTMENT
- ❖ DOCTOR 1:N APPOINTMENT
- ❖ DEPARTMENT 1:N DOCTOR
- ❖ PATIENT 1:N PATIENT\_RECORD
- ❖ MEDICINE 1:N PATIENT\_RECORD
- ❖ TREATMENT 1:N PATIENT\_RECORD
- ❖ DOCTOR 1:N PATIENT\_RECORD





# Many to Many Relationships

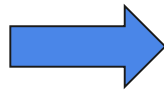
## N : M

A many-to-many relationship is a type of cardinality that refers to the relationship between two entities A and B in which A may contain a parent instance for which there are many children in B and vice versa. For example, think of A as Classes, and B as Students. A Class can have several Students, and a Student can be registered in several Classes.

Due to the fact that many medicines can be prescribed to many treatments, it was introduced the additional entity MEDICINE\_TREATMENT. The same occurs with PATIENT\_TREATMENT.



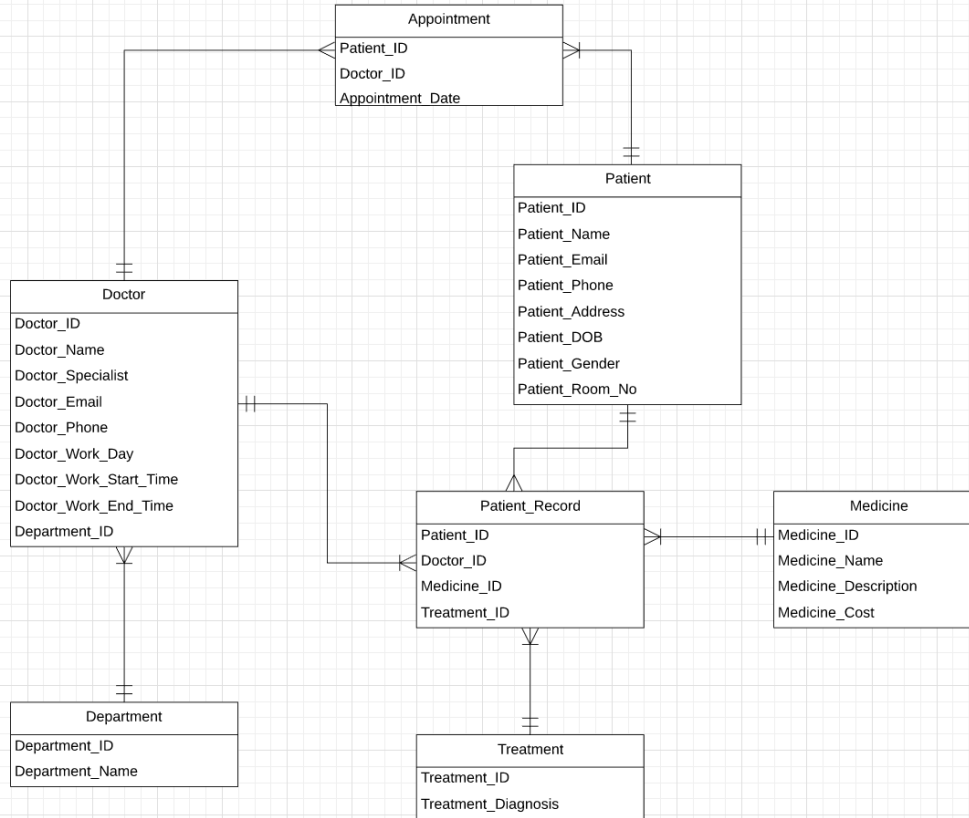
- ❖ PATIENT N:M TREATMENT
- ❖ MEDICINE N:M TREATMENT
- ❖ DOCTOR N:M TREATMENT



- ❖ PATIENT 1:N PATIENT\_RECORD
- ❖ MEDICINE 1:N PATIENT\_RECORD
- ❖ TREATMENT 1:N PATIENT\_RECORD
- ❖ DOCTOR 1:N PATIENT\_RECORD

## Hospital Management System

Ligia Calvão | Meenal Sawant | Rawaf Alselhi



Entity Relationship Model Diagram



# Relational Model








The Relational database represents the relation between tables. Primary key uniquely identifies the records the row of that table. Foreign key of one table is a primary of another table. Referential integrity constraints is based on the concept of Foreign Keys. Referential integrity constraint ensures that the values of one column in a table are valid based on values in another table. However, that key element in child table (FK) must exist in the parent table(PK). The advantage of the relational model are insert, select, modify and delete these operations can be performed in Relational Model and it minimizes the data redundancy issues.





# Tables and data types

## Tables

	Appointment
	Department
	Doctor
	Medicine
	Patient
	Patient_Record
	Treatment

Field Name	Data Type
Doctor_ID	AutoNumber
Doctor_Name	Short Text
Doctor_Specialist	Short Text
Doctor_Email	Short Text
Doctor_Phone	Short Text
Doctor_Work_Day	Short Text
Doctor_Work_Start_Time	Date/Time
Doctor_Work_End_Time	Date/Time
Department_ID	Number



# Relational Design

**Patient**(Patient\_ID(PK), Patient\_Name, Patient\_Email, Patient\_Phone, Patient\_Address, Patient\_DOB, Patient\_Gender, Patient\_Room\_No)

**Appointment**(Patient\_ID(PK, FK), Doctor\_ID(PK, FK), Appointment\_Date)

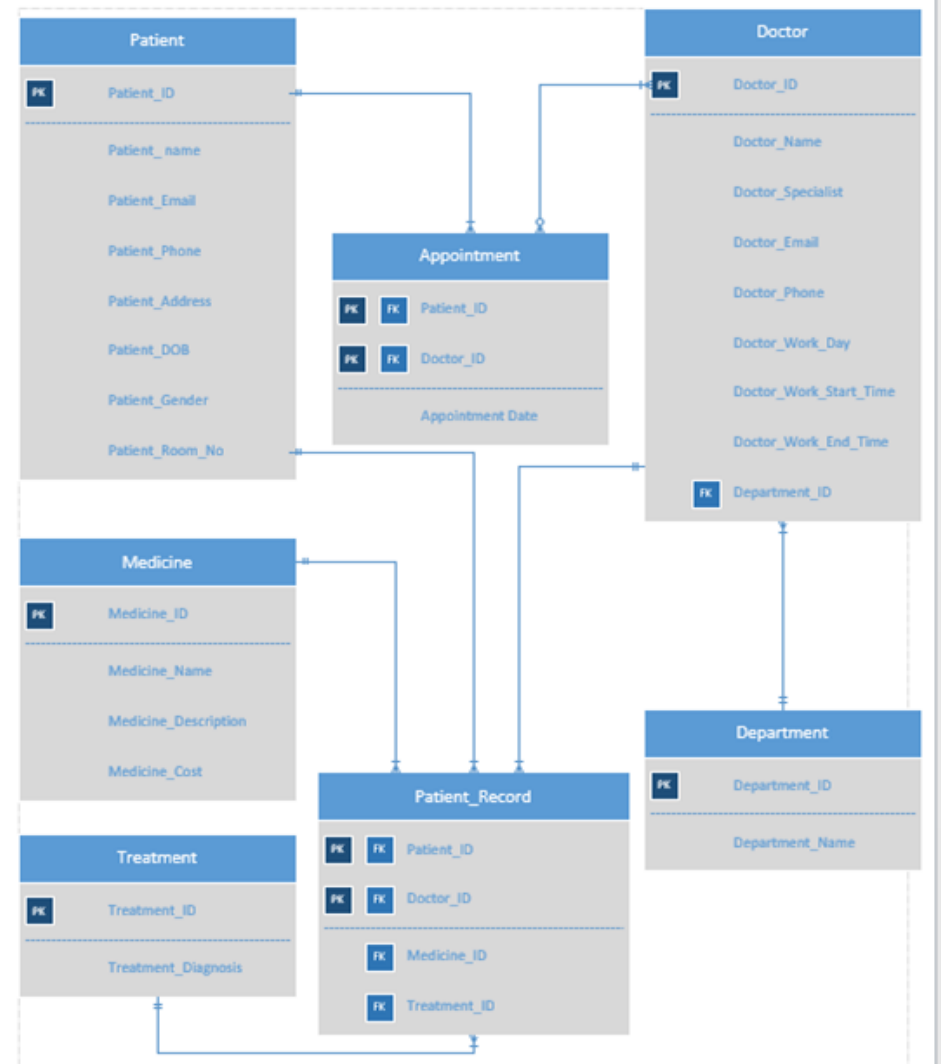
**Doctor**(Doctor\_ID(PK), Doctor\_Name, Doctor\_Specialist, Doctor\_Email, Doctor\_Phone, Doctor\_Work\_Day, Doctor\_Work\_Start\_Time, Doctor\_Work\_End\_Time, Department\_ID(FK))

**Department**(Department\_ID(PK), Department\_Name)

**Treatment**(Treatment\_ID(PK), Treatment\_Diagnosis)

**Medicine**(Medicine\_ID(PK), Medicine\_Name, Medicine\_Description, Medicine\_Cost)

**Patient\_Record**(Patient\_ID(PK, FK), Doctor\_ID(PK, FK), Medicine\_ID(FK), Treatment\_ID(FK))

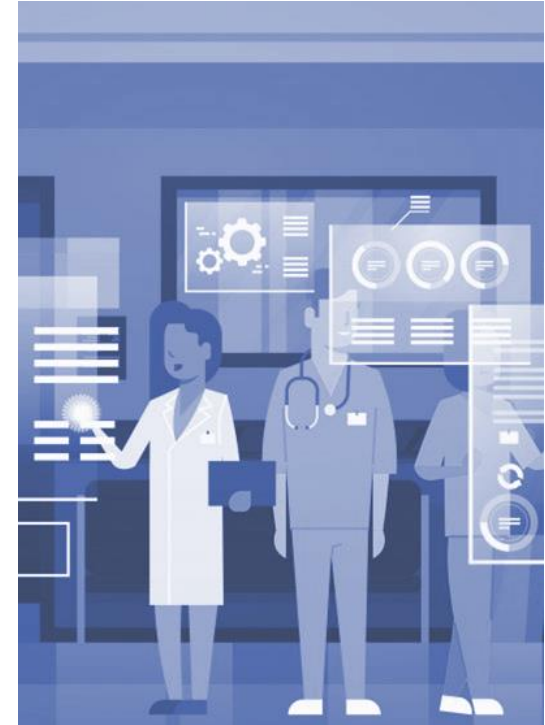


Logical design



# Constraints

For all the above relationships, We have enforced Referential Integrity to avoid the loss or inadvertent updating of data in the tables. Primary key in the table which are referenced in other tables as foreign keys have this integrity enforced. Also, we have applied Cascade Update, Cascade Delete so that whenever we update/delete the data in the parent table, the respective foreign key in the referenced table would get updated.







## Edit Relationships



Table/Query:	Related Table/Query:
Patient	Appointment
Patient_ID	Patient_ID

- ☒ Enforce Referential Integrity
- ☒ Cascade Update Related Fields
- ☒ Cascade Delete Related Records

Relationship Type: One-To-Many

OK

Cancel

Join Type..

Create New..

## Edit Relationships



Table/Query:	Related Table/Query:
Doctor	Appointment
Doctor_ID	Doctor_ID

- ☒ Enforce Referential Integrity
- ☒ Cascade Update Related Fields
- ☒ Cascade Delete Related Records

Relationship Type: One-To-Many

OK

Cancel

Join Type..

Create New..

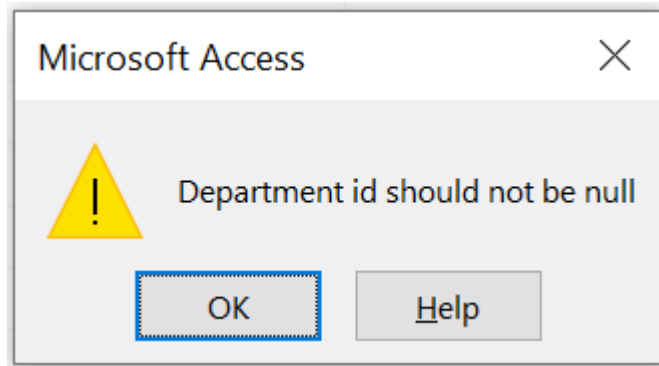
## Constraints



# Validation Rules

We have also added validation rule 'Is not NULL' for several fields in our tables where we wanted to enforce no null values for the field.

Example: Department\_ID in the Doctor table



	Field Name	Data Type
	Doctor_ID	AutoNumber
	Doctor_Name	Short Text
	Doctor_Specialist	Short Text
	Doctor_Email	Short Text
	Doctor_Phone	Short Text
	Doctor_Work_Day	Short Text
	Doctor_Work_Start_Time	Date/Time
	Doctor_Work_End_Time	Date/Time
	Department_ID	Number

General	Lookup
Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	0
Validation Rule	Is Not Null
Validation Text	Department_ID should not be null
Required	No
Indexed	Yes (Duplicates OK)
Text Align	General

**Validation Rule: Is Not Null**

# Queries

1 SQL: Number of doctors who their specialist is either “ General Physician” or “Dentist”:

SELECT \*

FROM DOCTOR

WHERE Doctor\_SPECIALIST IN( 'General Physician','Dentist');

Result:

Doctor_ID	Doctor_Name	Doctor_Specialist	Doctor_Email	Doctor_Phone	Doctor_Work_Day	Doctor_Work_Start_Ti	Doctor_Work_End_Time	Department_ID
4	Sophia	General Physician		6564434251	Thursday	9:00:00 AM	4:00:00 PM	5
7	Nicole	General Physician		6074545522	Monday	9:00:00 AM	4:00:00 PM	5
1	Harry	Dentist		8761232311	Monday	9:00:00 AM	4:00:00 PM	2
*	(New)							0

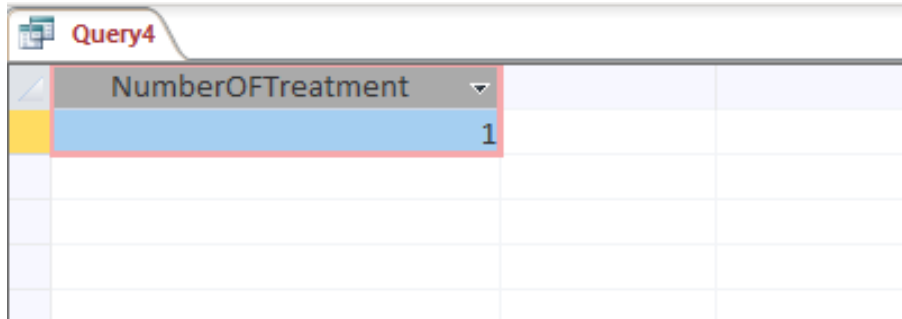


# Queries

2 SQL: Number of patients who got a root canal treatment:

```
SELECT COUNT(patient_ID) AS NumberOfTreatment  
FROM Patient_Treatment  
WHERE Treatment_ID = 1;
```

 Result:



NumberOfTreatment		
1		



# Queries

3 SQL: Find all Doctors who in the Pediatric department:

```
SELECT Doctor.Doctor_Name, Department.Department_name  
FROM Doctor INNER JOIN Department ON  
Doctor.Department_ID = Department.Department_ID  
WHERE Department.Department_Name = "Pediatrics";
```

 Result:

Patient_Record		Query1		Department		Doctor	
Doctor_Name		Department					
Jacob		Pediatrics					
*							



# Example Report



## List of Patients

Tuesday, December 10, 2019

2:56:39 PM

Patient_Name	Patient_Email	Patient_Address
Sam	sam@gmail.com	917 Penn Dr.Painesville, NY 44077
Nancy	Nancy@gmail.com	31, one drive, 134th street, NY 01267
Sarah	sarah@gmail.com	697 Primrose Ave. Gainesville, NY20155
Linda	linda@gmail.com	411 Ridge Dr.Santa Monica, NY 90403
Scott	scott@gmail.com	431 George Road Chaska, NY 55318

A photograph of a long, brightly lit hospital hallway. The hallway has white walls, a polished floor reflecting the overhead lights, and several doors on the right side. In the distance, medical equipment and a person can be seen. A blue semi-transparent banner is overlaid across the middle of the image.

# Thank You