**SVKM’s NMIMS**

**Mukesh Patel School of Technology Management & Engineering (Mumbai Campus)**

**Computer Engineering Department (B Tech CSE/CSBS Sem IV/BTI Sem VIII/MBA.Tech-IV)**

**Database Management System**

**Project Report**

|  |  |  |
| --- | --- | --- |
| Program | **MBA TECH CE** | |
| Semester | **4** | |
| Name of the Project: | **APPOINTMENT BOOKING AND DOCTOR’S REGISTRATION** | |
|  | | |
| Details of Project Members |  |  |
| Batch | Roll No. | Name |
| **A1** | **A234** | **AAYESHA PATEL** |
| **A1** | **A232** | **VIDHI PANCHAL** |
| **A1** | **A220** | **ATHARV AMIT DESHPANDE** |
| Date of Submission: **28 MARCH 2025** | | |

**Contribution of each project Members:**

|  |  |  |
| --- | --- | --- |
| **Roll No.** | **Name:** | **Contribution** |
| A234 | AAYESHA PATEL | CREATED DOCTOR’S FORM AND TABLE |
| A232 | VIDHI PANCHAL | CREATED DEPARTMENTS AND PATIENTS TABLE |
| A220 | ATHARV AMIT DESHPANDE | CREATED CHATBOT AND LINKING WITH DATABASE |

**Github link of your project:**

**Note:**

1. Create a readme file if you have multiple files
2. All files must be properly named (Example:R004\_DBMSProject)
3. Submit all relevant files of your work ( Report, all SQL files, Any other files)
4. **Plagiarism is highly discouraged (Your report will be checked for plagiarism)**

**Rubrics for the Project evaluation:**

|  |  |
| --- | --- |
| First phase of evaluation:  Innovative Ideas (5 Marks)  Design and Partial implementation (5 Marks) | 10 marks |
| Final phase of evaluation  Implementation, presentation and viva, Self-Learning and Learning Beyond classroom | 10 marks |

**Project Report**

**APPOINTMENT BOOKING AND DOCTOR’S REGISTRATION**

**by**

**AAYESHA, Roll number: A234**

**VIDHI , Roll number: A232**

**ATHARV , Roll number: A220**

**Course: DBMS**

**AY: 2024-25**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Sr no.** | **Topic** | **Page no.** |
| **1** | Storyline | 5 |
| **2** | Components of Database Design | 5 |
| **3** | Entity Relationship Diagram | 5 |
| **4** | Relational Model | 6 |
| **5** | Normalization | 6 |
| **6** | SQL Queries | 6 |
| **7** | Learning from the Project | 7 |
| **8** | Project Demonstration | 7 |
| **9** | Self-learning beyond classroom | 7 |
| **10** | Learning from the project | 8 |
| **8** | Challenges faced | 8 |
| **9** | Conclusion | 8 |

**I. Storyline**

We are working on making appointment booking and registration of doctors very easy in hospitals. We have created the hospital\_management database that keeps information about the doctors, patients, and departments. It is structured in a way that allows easy access to data while keeping track of how different pieces of data relate to each other. And Also This Project We Will Integrate it With Our Website which we’re Learning In Another Subject, That Allows us to Learn Database Connectivity So This Project Will Be More Practical & Insightful.

**II. Components of Database Design**

**Components of database:**

The Patient, Doctor, hospital & Department table consists of the following fields –

DOCTOR (doctor\_id, full\_name, specialization, department\_id, gender, date\_of\_birth, phone\_number, email, location, license\_number, experience)

PATIENT(patient\_id, name, hospital\_name, location, time, doctor\_id)

DEPARTMENT(department\_id, department\_name, department\_location, department\_full\_name)

HOSPITAL(hospital\_name, hospital\_id, location, type)

**Relationships:**

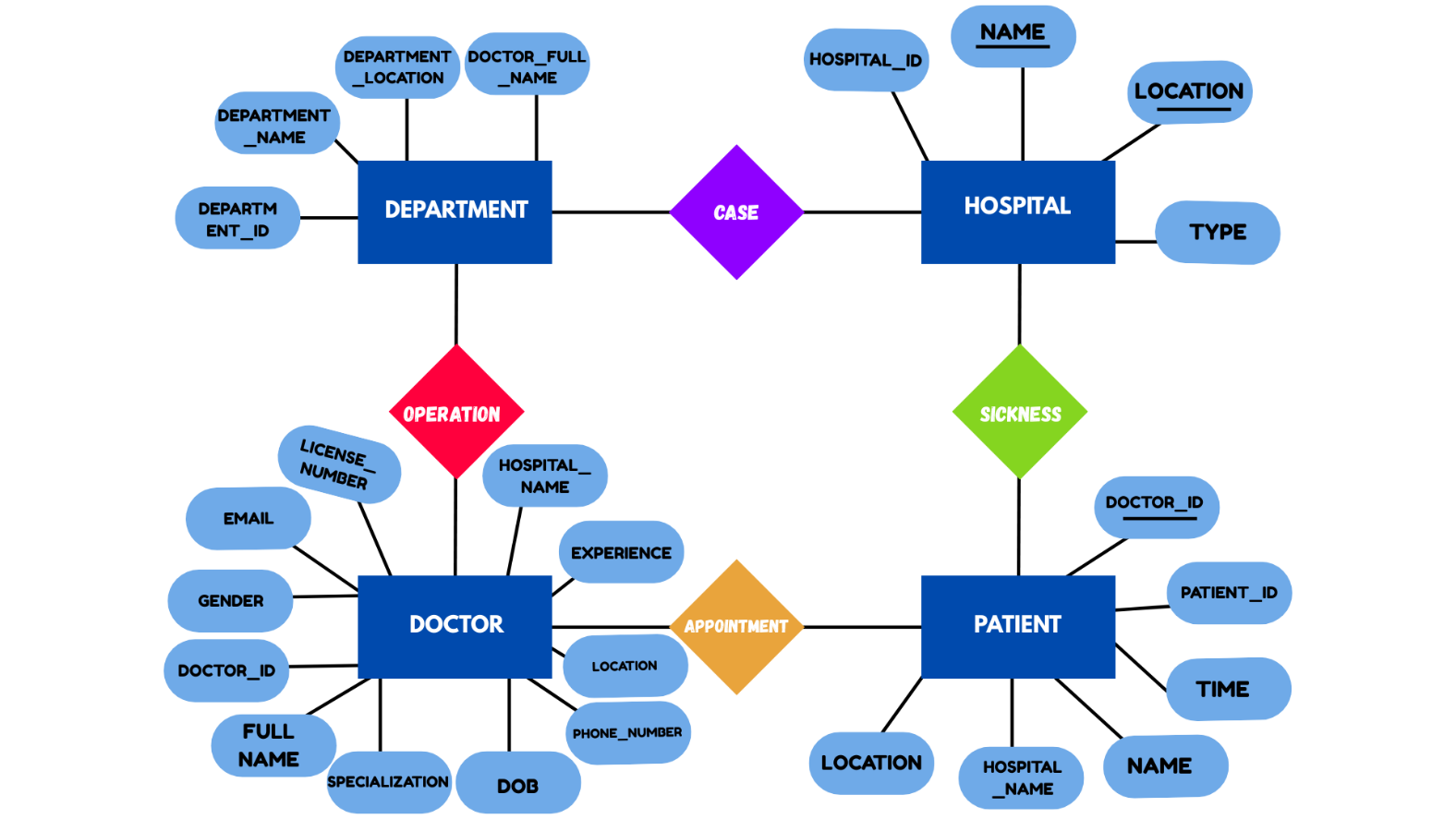
A department has many doctors (one-to-many).

**Many-To-One Relationship**: A patient books an appointment with a doctor (many-to-one).

**III. Entity Relationship Diagram**

Draw the ER diagram here. An example is shown:

You can also use software for drawing ER diagram



**IV. Relational Model**

The relational schema is as follows:

DOCTOR (doctor\_id PK, full\_name, specialization, department\_id FK, gender, date\_of\_birth, phone\_number, email, location, license\_number, experience)

PATIENT(patient\_id PK, name, hospital\_name FK, location, time, doctor\_id FK)

DEPARTMENT(department\_id PK, department\_name, department\_location, department\_full\_name)

HOSPITAL(hospital\_name, hospital\_id PK, location, type)

**V. Normalization**

Perform normalization (1NF, 2NF, 3NF, BCNF) as applicable for the entire database.

**VI. SQL Queries**

Using a DBMS software (SQLite3 or MySQL or any other of your choice):

* Create the tables

Departments Table:-

CREATE TABLE `departments` (

`department\_id` int NOT NULL AUTO\_INCREMENT,

`department\_name` varchar(100) DEFAULT NULL,

`department\_location` varchar(200) DEFAULT NULL,

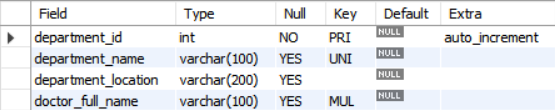
`doctor\_full\_name` varchar(100) DEFAULT NULL,

PRIMARY KEY (`department\_id`),

UNIQUE KEY `department\_name` (`department\_name`),

KEY `fk\_doctor\_name` (`doctor\_full\_name`)

)



Doctor’s Table:-

CREATE TABLE `doctor` (

`doctor\_id` int NOT NULL AUTO\_INCREMENT,

`full\_name` varchar(255) NOT NULL,

`date\_of\_birth` date DEFAULT NULL,

`gender` enum('Male','Female','Other') DEFAULT NULL,

`email` varchar(255) DEFAULT NULL,

`phone\_number` varchar(15) DEFAULT NULL,

`specialization` varchar(255) DEFAULT NULL,

`experience` int DEFAULT NULL,

`license\_number` varchar(50) DEFAULT NULL,

`hospital\_name` varchar(255) DEFAULT NULL,

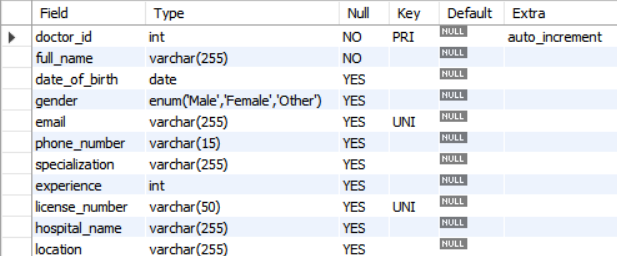
`location` varchar(255) DEFAULT NULL,

PRIMARY KEY (`doctor\_id`),

UNIQUE KEY `email` (`email`),

UNIQUE KEY `license\_number` (`license\_number`)

)



Patient’s Table:-

CREATE TABLE patients (

patient\_id INT NOT NULL AUTO\_INCREMENT,

name VARCHAR(100) NOT NULL,

doctor\_id INT,

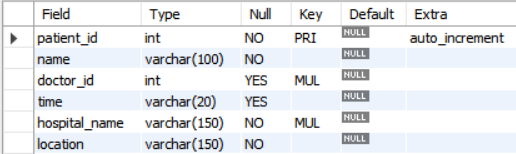
department\_name VARCHAR(100),

PRIMARY KEY (patient\_id),

FOREIGN KEY (doctor\_id) REFERENCES doctor(doctor\_id),

FOREIGN KEY (department\_name) REFERENCES departments(department\_name)

)



Hospital’s Table:-

CREATE TABLE `hospital` (

`id` int NOT NULL AUTO\_INCREMENT,

`name` varchar(255) NOT NULL,

`location` varchar(255) NOT NULL,

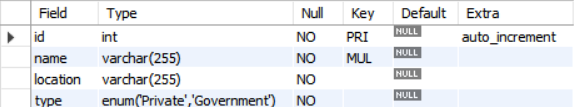
`type` enum('Private','Government') NOT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `unique\_hospital` (`name`,`location`),

UNIQUE KEY `hospital\_unique` (`name`,`location`)

)



* Populate the tables (insert some meaningful data, at least 10 tuples for each relation)

Department’s Table:-

INSERT INTO departments (department\_name, department\_location, doctor\_full\_name)

VALUES

('Neurology', 'Navi Mumbai', 'Meera Joshi'),

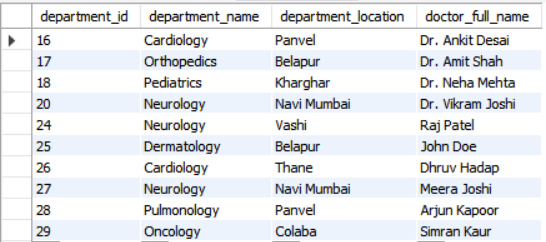
('Pulmonology', 'Panvel', 'Arjun Kapoor'),

('Oncology', 'Colaba', 'Simran Kaur'),

('Neurology', 'Vashi', 'Raj Patel'),

('Dermatology', 'Belapur', 'John Doe'),

('Cardiology', 'Thane','Dhruv Hadap');



* Run SQL queries (minimum 20) covering **all concepts** learned in the class

This section should contain the question, SQL code, and the output snapshot for each query.

**VI. Project demonstration**

* Tools/software/ libraries used: We used MySQL workbench software
* Integration: Connecting database to a website and chatbot to allow real-time appointments bookings and doctor’s admission.
* Screenshot and Description of the Demonstration of project ( If GUI is made)

A screenshot of a computer

AI-generated content may be incorrect.

**VII. Self -Learning beyond classroom**

:

* What new aspects did you learn on your own ? You have to mention learning beyond the classroom

As a result, all of the self-learning in the absence of the teacher

Here is what I explored through this project:

* Web applications that require connectivity to a database
* Best practice for SQL query to performance boost.
* Data Security: ensure to be responsible with handling sensitive patient information.

**VIII. Learning from the Project**

* How this project helped you?

It gave me a deeper insight into relationships and normalization of the database design. Besides, it was a great experience to get some hands-on experience with database connectivity and full-stack development, by integrating it with a website.

**IX. Challenges Faced**

* Maintaining data integrity when we have several relationships
* Minimize all queries for efficiency.
* Handling the database connections for live updates.

**X. Conclusion**

* What are the key takeaways from the project?

This project sharpened our database concepts and opened my eyes to real world problems.