**Database Design**

**For**

**HealthCare Management System**

**1. Introduction to MySQL**

MySQL is an open-source relational database management system (RDBMS) that uses structured query language (SQL) to manage and manipulate data in a database. It is widely used for various applications, from small web applications to large enterprise systems.

MySQL's key features include:

* Scalability: Capable of handling large amounts of data and concurrent connections.
* Flexibility: Supports various data types and storage engines.
* Performance: Optimized for speed and efficiency.
* Reliability: Known for its stability and robustness.

**2. Installation of MySQL**

MySQL can be installed on various operating systems, including Windows, macOS, and Linux. Here are the general steps to install MySQL:

**Windows:**

* Download the MySQL installer from the official website.

<https://dev.mysql.com/downloads/installer/>

* Run the installer and follow the on-screen instructions.
* Choose the installation type (Typical, Complete, or Custom). Recommended Custom.
* Set a root password for the MySQL server.

**3. E-R Diagram (ERD)**

An Entity-Relationship Diagram (ERD) is a visual representation of the data model that shows the entities, attributes, relationships between entities, and cardinality. ERDs are commonly used in database design to help developers and stakeholders understand the structure and relationships within a database.

**Identify Entities**

* Start by identifying the main entities in your system. These are the objects or concepts about which you want to store data.
* Each entity should correspond to a table in your database.

**Define Attributes**

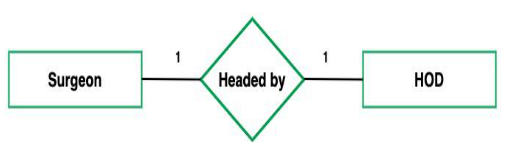
* For each entity, list the attributes (properties or fields) that describe it.
* These attributes will become columns in the corresponding database table.

**Identify Relationships**

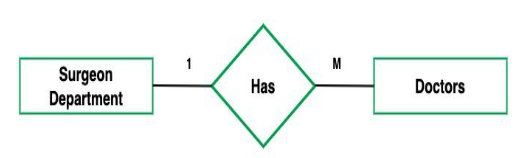
* Determine how entities are related to each other. There are three types of relationships: one-to-one (1:1), one-to-many (1:N), and many-to-many (N:M).
* Represent these relationships using lines connecting the entities.

Let’s see a few examples of relationships:

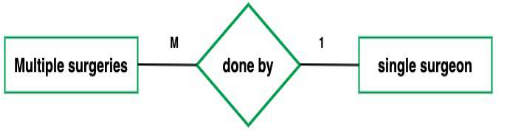
**One to One**



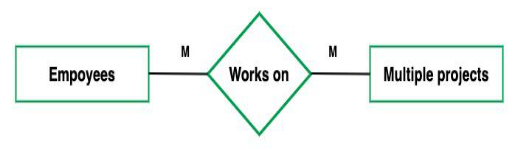
**One to Many**



**Many to One**



**Many to Many**



**Cardinality Notation**

Cardinality represents the number of times an entity of an entity set participates in a relationship set. Or we can say that the cardinality of a relationship is the number of tuples (rows) in a relationship.

* Use notation (such as Crow's Foot Notation or Chen Notation) to indicate the cardinality of each relationship.
* Cardinality describes how many instances of one entity are related to how many instances of another entity.
* Common notations include:
* One (1)
* Zero or one (0..1)
* Many (N)
* Zero or many (0..N)

**Optional:**

**Add Attributes and Constraints**

* Include additional information in your ERD, such as primary keys, foreign keys, and constraints (e.g., unique constraints).

**Create the Diagram**

* Use specialized diagramming software or tools (e.g., Lucidchart, draw.io, or even pen and paper) to create your ERD.

**Refine and Review:**

* Review your ERD with stakeholders and team members to ensure it accurately represents the data model and relationships. Make any necessary refinements.

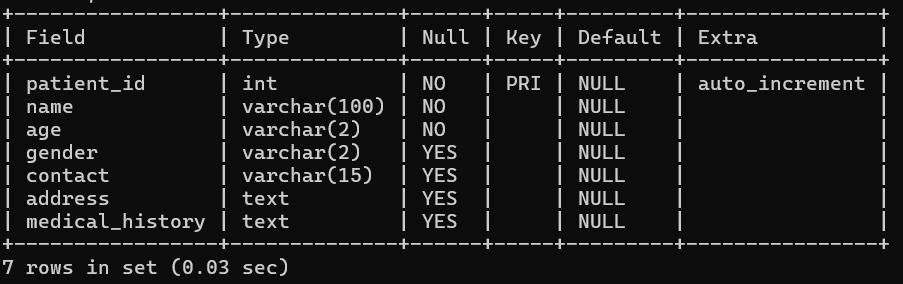
The primary entities in the Healthcare Management System are:

1. **Patient**
   * **Attributes**: patient\_id (Primary Key), name, contact, address, medical\_history.
   * **Relationships**:
     + One-to-many with **Appointment** (a patient can have multiple appointments).
     + One-to-many with **Prescription** (a patient can have multiple prescriptions).
     + One-to-many with **Diagnosis** (a patient can have multiple diagnoses).
     + One-to-many with **LabTest** (a patient can have multiple lab tests).
     + One-to-one with **TreatmentPlan** (each patient has one treatment plan).
     + One-to-one with **Insurance** (each patient may have one insurance policy).
2. **Doctor**
   * **Attributes**: doctor\_id (Primary Key), name, specialization, contact.
   * **Relationships**:
     + One-to-many with **Appointment** (a doctor can have multiple appointments).
     + Many-to-many with **Patient** through **Appointment** (each doctor can see multiple patients, and each patient can see multiple doctors).
     + Many-to-one with **Department** (a doctor belongs to one department).
3. **Nurse**
   * **Attributes**: nurse\_id (Primary Key), name, contact, department\_id (Foreign Key).
   * **Relationships**:
     + Many-to-one with **Department** (a nurse is assigned to one department).
4. **Receptionist**
   * **Attributes**: receptionist\_id (Primary Key), name, contact.
   * **Relationships**:
     + One-to-many with **Appointment** (a receptionist manages multiple appointments).
5. **Appointment**
   * **Attributes**: appointment\_id (Primary Key), patient\_id (Foreign Key), doctor\_id (Foreign Key), date\_time.
   * **Relationships**:
     + Many-to-one with **Patient**.
     + Many-to-one with **Doctor**.
     + One-to-one with **Prescription** (each appointment may have one prescription).
6. **Prescription**
   * **Attributes**: prescription\_id (Primary Key), appointment\_id (Foreign Key), medicine\_name, dosage, duration.
   * **Relationships**:
     + Many-to-one with **Appointment**.
7. **Medicine**
   * **Attributes**: medicine\_id (Primary Key), name, quantity, expiry\_date.
   * **Relationships**:
     + Many-to-many with **Prescription** (a prescription can contain multiple medicines, and a medicine can be prescribed in multiple prescriptions).
8. **Billing**
   * **Attributes**: billing\_id (Primary Key), patient\_id (Foreign Key), amount, date.
   * **Relationships**:
     + Many-to-one with **Patient**.
     + One-to-many with **Payment** (a bill may be paid in multiple payments).
9. **Payment**
   * **Attributes**: payment\_id (Primary Key), billing\_id (Foreign Key), method, amount.
   * **Relationships**:
     + Many-to-one with **Billing**.
10. **Room**
    * **Attributes**: room\_id (Primary Key), type, availability.
    * **Relationships**:
      + One-to-one with **Patient** (each room is assigned to one patient at a time).
11. **Department**
    * **Attributes**: department\_id (Primary Key), name, head\_doctor (Foreign Key to Doctor).
    * **Relationships**:
      + One-to-many with **Doctor** and **Nurse**.
12. **TreatmentPlan**
    * **Attributes**: plan\_id (Primary Key), patient\_id (Foreign Key), description.
    * **Relationships**:
      + One-to-one with **Patient**.
13. **Insurance**
    * **Attributes**: insurance\_id (Primary Key), patient\_id (Foreign Key), company, policy\_number.
    * **Relationships**:
      + One-to-one with **Patient**.

**Table Structure**

Here are some of the tables from the system to show the basic structure of the structure.

**Patient:**



**Doctor:**

A screen shot of a computer screen

Description automatically generated

**Nurse:**

A black screen with white text

Description automatically generated

**Appointment:**

A black screen with white text

Description automatically generated

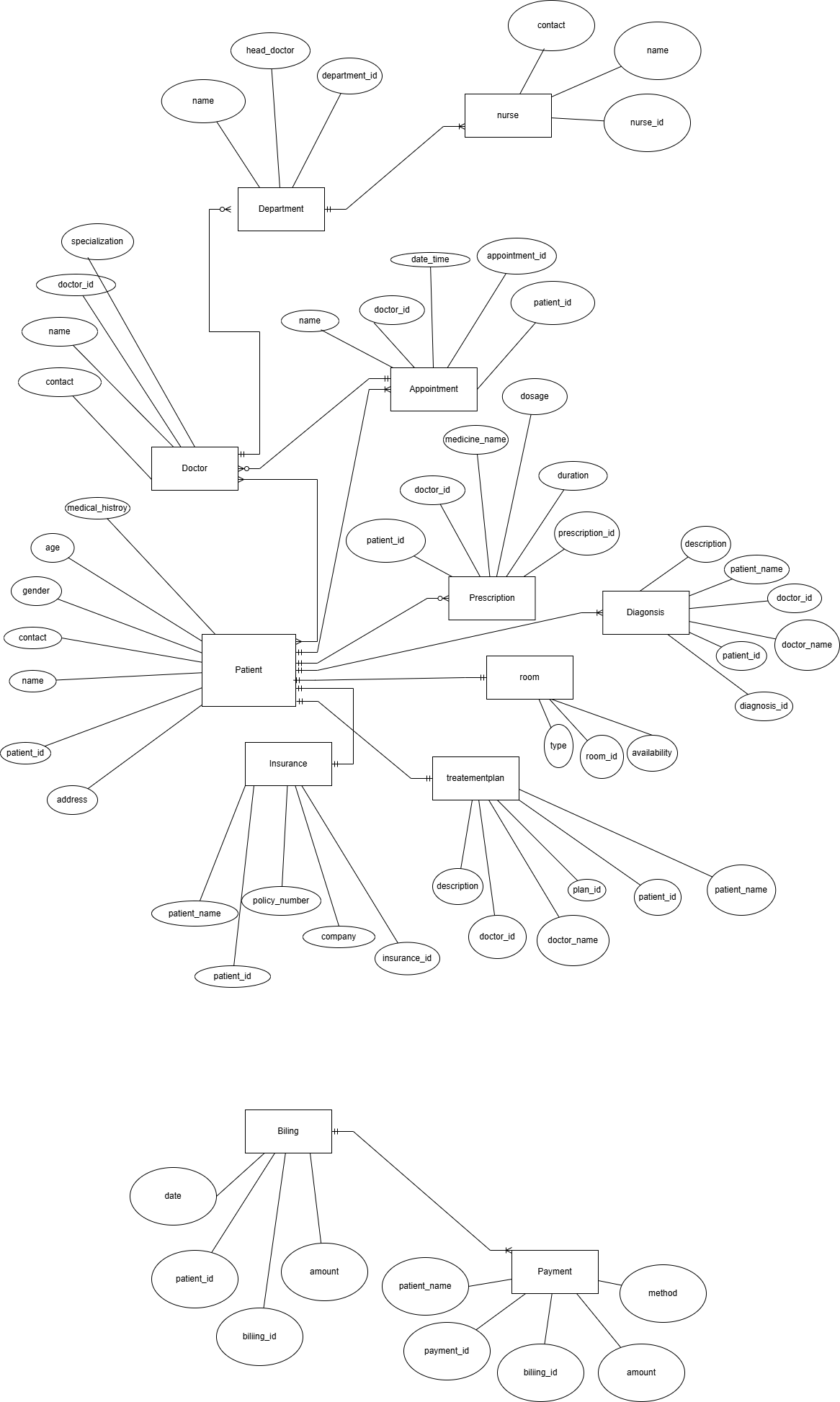
**Prescription:**

A screen shot of a computer

Description automatically generated

**ERD Diagram**

Now, let’s create the ER diagram to visually represent the entities and relationships.



**4. Creating a Database**

Using MySQL server, create a new database for your student management system. You can do this with SQL commands or through the graphical interface.

*CREATE DATABASE HMS;*

**5. Using a Database**

Before performing any operations on a database, you need to select it using the USE statement:

*USE HMS;*

**6. Creating the tables for each entity**

*Use HMS;*

*-- Patient Table*

*CREATE TABLE Patient (*

*patient\_id INT PRIMARY KEY AUTO\_INCREMENT,*

*name VARCHAR(100) NOT NULL,*

*age VARCHAR(2) NOT NULL,*

*gender VARCHAR(2),*

*contact VARCHAR(15),*

*address TEXT,*

*medical\_history TEXT*

*);*

*-- Doctor Table*

*CREATE TABLE Doctor (*

*doctor\_id INT PRIMARY KEY AUTO\_INCREMENT,*

*name VARCHAR(100) NOT NULL,*

*specialization VARCHAR(50),*

*contact VARCHAR(15)*

*);*

*-- Nurse Table*

*CREATE TABLE Nurse (*

*nurse\_id INT PRIMARY KEY AUTO\_INCREMENT,*

*name VARCHAR(100) NOT NULL,*

*contact VARCHAR(15),*

*department\_id INT*

*);*

*-- Receptionist Table*

*CREATE TABLE Receptionist (*

*receptionist\_id INT PRIMARY KEY AUTO\_INCREMENT,*

*name VARCHAR(100) NOT NULL,*

*contact VARCHAR(15)*

*);*

*-- Appointment Table*

*CREATE TABLE Appointment (*

*appointment\_id INT PRIMARY KEY AUTO\_INCREMENT,*

*patient\_id INT,*

*doctor\_id INT,*

*date\_time DATETIME,*

*FOREIGN KEY (patient\_id)*

*REFERENCES Patient (patient\_id),*

*FOREIGN KEY (doctor\_id)*

*REFERENCES Doctor (doctor\_id)*

*);*

*-- Prescription Table*

*CREATE TABLE Prescription (*

*prescription\_id INT PRIMARY KEY AUTO\_INCREMENT,*

*patient\_id INT,*

*doctor\_id INT,*

*medicine\_name VARCHAR(100),*

*dosage VARCHAR(50),*

*duration VARCHAR(50),*

*FOREIGN KEY (patient\_id)*

*REFERENCES Patient (patient\_id),*

*FOREIGN KEY (doctor\_id)*

*REFERENCES Doctor (doctor\_id)*

*);*

*-- Medicine Table*

*CREATE TABLE Medicine (*

*medicine\_id INT PRIMARY KEY AUTO\_INCREMENT,*

*name VARCHAR(100) NOT NULL,*

*quantity INT,*

*expiry\_date DATE,*

*price DECIMAL(10 , 2 )*

*);*

*-- Diagnosis Table*

*CREATE TABLE Diagnosis (*

*diagnosis\_id INT PRIMARY KEY AUTO\_INCREMENT,*

*patient\_id INT,*

*patient\_name VARCHAR(100) NOT NULL,*

*doctor\_id INT,*

*doctor\_name VARCHAR(100) NOT NULL,*

*description TEXT,*

*FOREIGN KEY (patient\_id)*

*REFERENCES Patient (patient\_id)*

*);*

*-- LabTest Table*

*CREATE TABLE LabTest (*

*test\_id INT PRIMARY KEY AUTO\_INCREMENT,*

*patient\_id INT,*

*patient\_name VARCHAR(100) NOT NULL,*

*test\_type VARCHAR(100),*

*result VARCHAR(100),*

*FOREIGN KEY (patient\_id)*

*REFERENCES Patient (patient\_id)*

*);*

*-- Billing Table*

*CREATE TABLE Billing (*

*billing\_id INT PRIMARY KEY AUTO\_INCREMENT,*

*patient\_id INT,*

*amount DECIMAL(10 , 2 ),*

*date DATE,*

*FOREIGN KEY (patient\_id)*

*REFERENCES Patient (patient\_id)*

*);*

*-- Payment Table*

*CREATE TABLE Payment (*

*payment\_id INT PRIMARY KEY AUTO\_INCREMENT,*

*billing\_id INT,*

*patient\_name VARCHAR(100) NOT NULL,*

*method VARCHAR(50),*

*amount DECIMAL(10 , 2 ),*

*FOREIGN KEY (billing\_id)*

*REFERENCES Billing (billing\_id)*

*);*

*-- Room Table*

*CREATE TABLE Room (*

*room\_id INT PRIMARY KEY AUTO\_INCREMENT,*

*type VARCHAR(50),*

*availability BOOLEAN DEFAULT TRUE*

*);*

*-- TreatmentPlan Table*

*CREATE TABLE TreatmentPlan (*

*plan\_id INT PRIMARY KEY AUTO\_INCREMENT,*

*patient\_id INT,*

*patient\_name VARCHAR(100) NOT NULL,*

*doctor\_id INT,*

*doctor\_name VARCHAR(100) NOT NULL,*

*description TEXT,*

*FOREIGN KEY (patient\_id)*

*REFERENCES Patient (patient\_id),*

*FOREIGN KEY (doctor\_id)*

*REFERENCES Doctor (doctor\_id)*

*);*

*-- Insurance Table*

*CREATE TABLE Insurance (*

*insurance\_id INT PRIMARY KEY AUTO\_INCREMENT,*

*patient\_id INT,*

*patient\_name VARCHAR(100) NOT NULL,*

*company VARCHAR(100),*

*policy\_number VARCHAR(50),*

*FOREIGN KEY (patient\_id)*

*REFERENCES Patient (patient\_id)*

*);*