**Database Design**

A patient care system is a crucial component of any healthcare organization, enabling efficient management of patient data, appointments, medicine, and payments. A well-designed database is essential to support this system, ensuring data integrity, consistency, and scalability. This document outlines the database design for a patient care system, comprising seven entities: Admin, Patient, Doctor, Appointment, Medicine, Prescription, and Payment.

It will help to manage the below functionalities.

Entities and their attributes:

1. **Person** (Parent Class)

* Person id (primary key)
* First name
* Last name
* dob
* gender
* address
* phone no
* Email id
* Password

1. **Patient** (Child Class)

* Patient id (Primary key)
* Symptoms

1. **Doctor** (Child Class)

* Doctor id (Primary key)
* specialization
* schedule

1. **Staff** (Child Class)

* Staff id (primary key)
* Department

1. **Appointment**

* Appointment id (Primary key)
* appointment date
* status
* patient id (Foreign key)
* doctor id (Foreign key)

1. **Medicine**

* Medicine id (Primary Key)
* name
* description
* price

1. **Prescription**

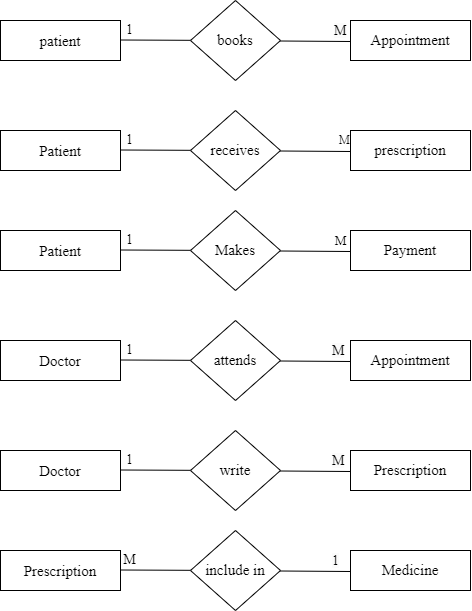
* Prescription id (primary key)
* dose
* duration
* patient id (foreign key)
* doctor id (foreign key)
* medicine id (foreign key)

1. **Payment**

* Payment id (primary key)
* amount
* payment date
* payment method
* patient id (foreign key)

**Identifying Relationship:**

* 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.

****

**Relationships:**

The relationships between the Admin, Patient, Doctor, Appointment, Medicine, Prescription, and Payment tables in a patient care system database design are as follows:

**Admin:**

Attributes:

* Admin id (primary key)
* First name
* Last name
* Address
* Phone no
* Email Id
* password

Relationship:

* No relationships with other tables (Admin is a standalone entity)

**Patient:**

Attributes:

* Patient id (Primary key)
* First name
* Last name
* dob
* gender
* address
* phone no
* Email Id
* password

Relationship:

* A patient can have multiple appointments (one-to-many)
* A patient can have multiple prescriptions (one-to-many)
* A patient can make multiple payments (one-to-many)

**Doctor:**

Attributes:

* Doctor id (Primary key)
* First name
* Last name
* specialization
* schedule
* Phone no
* Email Id
* password

Relationship:

* A doctor can have multiple appointments (one-to-many)
* A doctor can prescribe multiple medicines (one-to-many)

**Appointment:**

Attributes:

* Appointment id (Primary key)
* appointment date
* status
* patient id (Foreign key)
* doctor id (Foreign key)

Relationship:

* An appointment is associated with one patient (many-to-one)
* An appointment is associated with one doctor (many-to-one)

**Medicine:**

Attributes:

* Medicine id (Primary Key)
* Name
* Description
* price

Relationship:

* A medicine can be prescribed to multiple patients (one-to-many)

**Prescription:**

Attributes:

* Prescription id (primary key)
* dose
* duration
* patient id (foreign key)
* doctor id (foreign key)
* medicine id (foreign key)

Relationship:

* A prescription is associated with one patient (many-to-one)
* A prescription is associated with one doctor (many-to-one)
* A prescription is associated with Many medicine (one-to-many)

**Payment:**

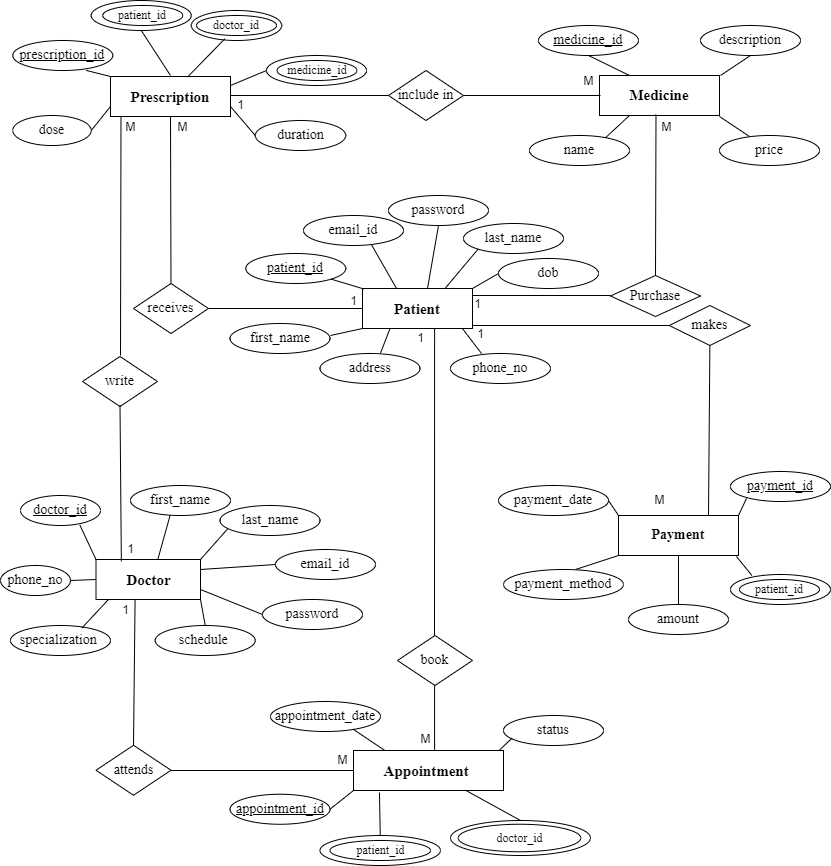
Attributes:

* Payment id (primary key)
* amount
* payment date
* payment method
* patient id (foreign key)

Relationship:

* A payment is associated with one patient (many-to-one)

**E-R Diagram (ERD)**

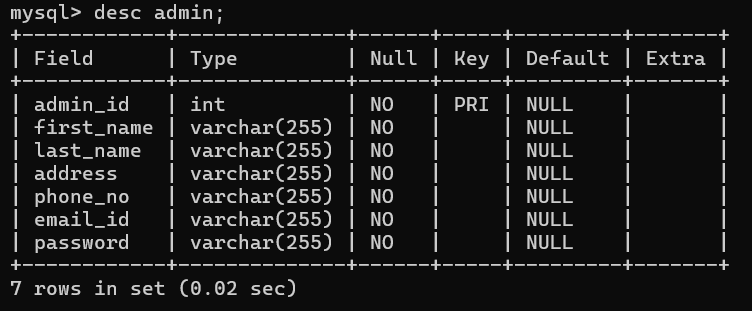
An Entity-Relationship Diagram (ERD) is a visual representation of the data model that shows the entities, attributes, relationships between entities, and cardinality. ERDs are commonly used in database design to help developers and stakeholders understand the structure and relationships within a database.

**In this ERD:**

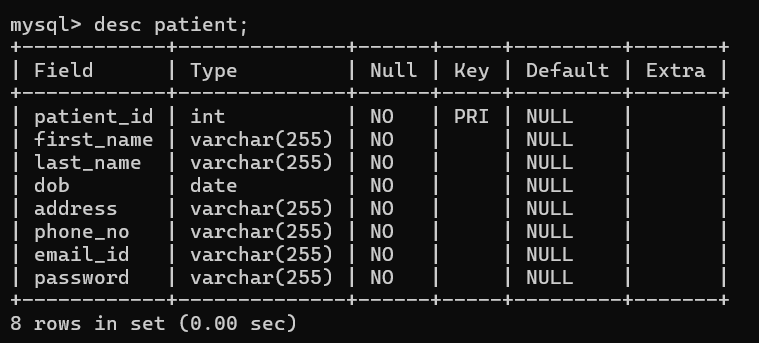
* patient can book appointment and each doctor has attends multiple appointments, creating a one-to-many relationship.
* The appointment entity serves as a bridge table between patient and doctor entities to represent this relationship.
* One patient can be receiving Many prescriptions included in multiple medicines (One-to-many relationship).
* Each patient can make multiple payments (one-to-many relationship).

**Table Structure:**

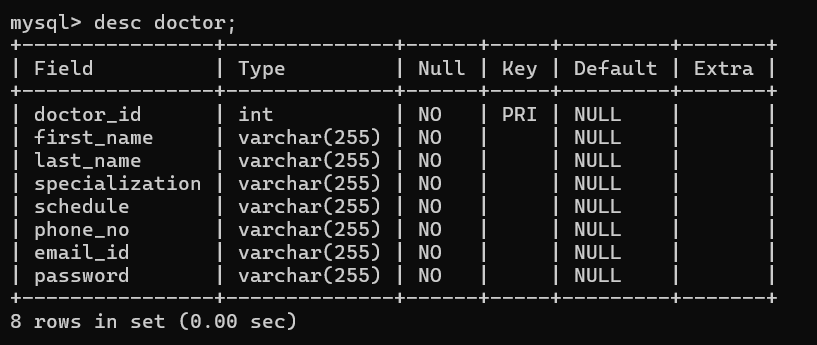
**Admin:**



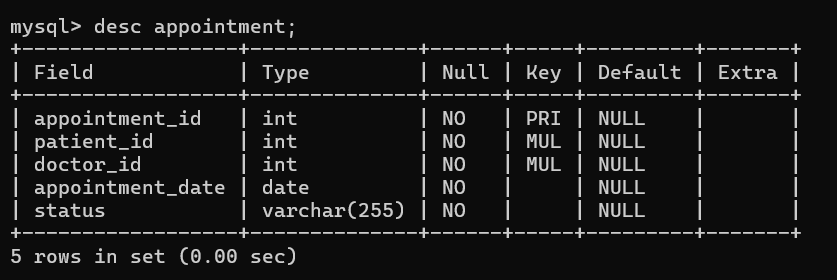
**Patient:**

****

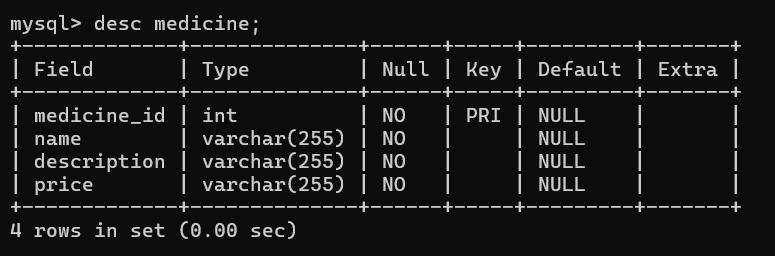
**Doctor:**

****

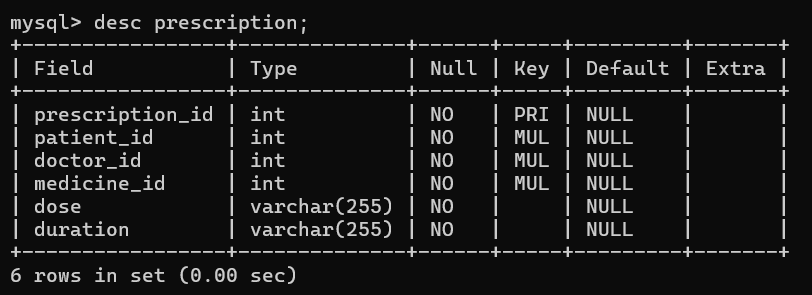
**Appointment:**

****

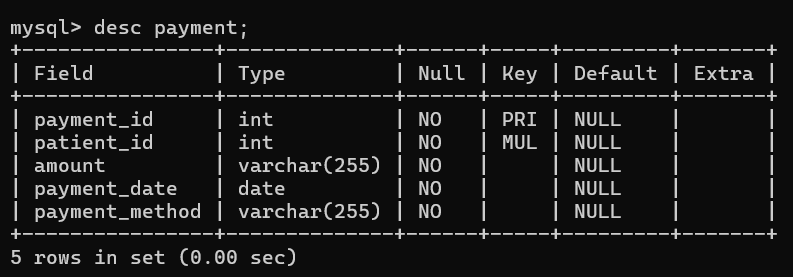
**Medicine:**

****

**Prescription:**

****

**Payment:**

****

**Creating a Database:**

Using MySQL server, create a new database for your Patient Care System. You can do this with SQL commands or through the graphical interface.

CREATE DATABASE patient\_care\_system;

**Using a Database:**

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

USE patient\_care\_system;

**Creating the tables for each entity:**

CREATE TABLE Admin (

admin\_id INT PRIMARY KEY,

first\_name VARCHAR(255) NOT NULL,

last\_name VARCHAR(255) NOT NULL,

address VARCHAR(255) NOT NULL,

phone\_no VARCHAR(255) NOT NULL,

email\_id VARCHAR(255) NOT NULL,

password VARCHAR(255) NOT NULL

);

CREATE TABLE Patient (

patient\_id INT PRIMARY KEY,

first\_name VARCHAR(255) NOT NULL,

last\_name VARCHAR(255) NOT NULL,

dob DATE NOT NULL,

address VARCHAR(255) NOT NULL,

phone\_no VARCHAR(255) NOT NULL,

email\_id VARCHAR(255) NOT NULL,

password VARCHAR(255) NOT NULL

);

CREATE TABLE Doctor (

doctor\_id INT PRIMARY KEY,

first\_name VARCHAR(255) NOT NULL,

last\_name VARCHAR(255) NOT NULL,

specialization VARCHAR(255) NOT NULL,

schedule VARCHAR(255) NOT NULL,

phone\_no VARCHAR(255) NOT NULL,

email\_id VARCHAR(255) NOT NULL,

password VARCHAR(255) NOT NULL

);

CREATE TABLE Appointment (

appointment\_id INT PRIMARY KEY,

patient\_id INT NOT NULL,

doctor\_id INT NOT NULL,

appointment\_date DATE NOT NULL,

status VARCHAR(255) NOT NULL,

FOREIGN KEY (patient\_id) REFERENCES Patient(patient\_id),

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

);

CREATE TABLE Medicine (

medicine\_id INT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

description VARCHAR(255) NOT NULL,

price VARCHAR(255) NOT NULL

);

CREATE TABLE Prescription (

prescription\_id INT PRIMARY KEY,

patient\_id INT NOT NULL,

doctor\_id INT NOT NULL,

medicine\_id INT NOT NULL,

dose VARCHAR(255) NOT NULL,

duration VARCHAR(255) NOT NULL,

FOREIGN KEY (patient\_id) REFERENCES Patient(patient\_id),

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

FOREIGN KEY (medicine\_id) REFERENCES Medicine(medicine\_id)

);

CREATE TABLE Payment (

payment\_id INT PRIMARY KEY,

patient\_id INT NOT NULL,

amount VARCHAR(255) NOT NULL,

payment\_date DATE NOT NULL,

payment\_method VARCHAR(255) NOT NULL,

FOREIGN KEY (patient\_id) REFERENCES Patient(patient\_id)

);

**Insert records**

Add data to your tables to work with. This step helps you test your database.

* **Insert admin**

INSERT INTO Admin(admin\_id, first\_name, last\_name, address, phone\_no, email\_id, password) VALUES

(1, 'Akanksha', 'Salunkhe', 'Satara', '9856325412', 'akanksha@gmail.com', 'Akanksha@123');

* **Insert patient**

INSERT INTO Patient (patient\_id, first\_name, last\_name, dob, address, phone\_no, email\_id, password)VALUES

(1, 'Sunny', 'Salunkhe', '2003-07-08', 'Pune', '8956895689', 'sunny@gmail.com', 'sunny@123'),

(2, 'Nivedita', 'Sonkawade', '2001-11-01', 'Solapur', '8787458745', 'nivedita@gmail.com', 'nivedita@123'),

(3, 'Shravani', 'Sawant', '2005-03-28', 'Satara', '3256325632', 'shravani@gmail.com', 'shravani@123'),

(4, 'Shriyank', 'Pawar', '2017-08-23', 'Satara', '1254564564', 'shriyank@gmail.com', 'shriyank@123'),

(5, 'Vijay', 'Dhage', '1996-10-24', 'Wai', '7598632145', 'vijay@gmail.com', 'vijay@123');

* **Insert doctor**

INSERT INTO Doctor (doctor\_id, first\_name, last\_name, specialization, schedule, phone\_no, email\_id, password)VALUES

(1, 'Dr. Ajay', 'Sawant', 'Aryurveda', '10am to 2pm','9999699969', 'ajay@gmail.com', 'ajay@123'),

(2, 'Dr. Neeta', 'Bhosale', 'Allopathic', '11am to 3pm', '8989563256', 'neeta@gmail.com', 'neeta@123'),

(3, 'Dr. Vikram', 'More', 'Dental', '9am to 1 pm', '7878454512', 'vikram@gmail.com', 'vikram@123'),

(4, 'Dr. Shriya', 'Pawar', 'Naturopathy', '10am to 2 pm', '8885649658', 'shriya@gmail.com', 'shriya@123'),

(5, 'Dr. Kalpita', 'Dhole', 'Homeopathy', '2pm to 7pm', '98563214785', 'kalpita@gmail.com', 'kalpita@123');

* **Insert appointment**

INSERT INTO Appointment(appointment\_id, appointment\_date, status, patient\_id, doctor\_id)VALUES

(1, '2024-06-04', 'booked', 1, 1),

(2, '2024-06-06', 'booked', 2, 3),

(3, '2024-06-07', 'booked', 3, 2);

* **Insert medicine**

INSERT INTO Medicine (medicine\_id, name, description, price) VALUES

(1, 'Dita', 'Skin UIcers, Fever, Increasing Lactation', 100),

(2, 'Antibiotics', 'pencillin, amoxicillin, vancomycin, augmentin', 200),

(3, 'ibuprofen', 'Advil, Nuprin, Motrin', 150),

(4, 'Paracetamol', 'headache, pain',100),

(5, 'crocin', 'toothache, musculoskeletal pain, migraine',120);

* **Insert prescription**

INSERT INTO Prescription (prescription\_id, dose, duration, patient\_id, doctor\_id, medicine\_id) VALUES

(1, '1st', '5 days', 1, 1, 1),

(2, '2nd', '2 days', 2, 2, 2),

(3, '1st', '15 days', 3, 3, 3);

* **Insert payment**

INSERT INTO Payment (payment\_id, amount, payment\_date, payment\_method, patient\_id) VALUES

(1, '1000', '2024-06-04', 'Google Pay', 1),

(2, '2000', '2024-06-06', 'Phone Pay', 2),

(3, '1500', '2024-06-07', 'Cash', 3);