A black and white logo

AI-generated content may be incorrect.

Hospital Management System

Saraf Abhinav Bala (23CSB0A33)

Venneti Teja Sree Charan (23CSB0A34)

18th April 2025

*A database management system for Hospital*The goal of this project is to **design and implement a Hospital Management System (HMS)** that:

* Provides secure and centralized management of patient and doctor records.
* Automates the registration process and assigns unique IDs to each patient.
* Allows users to book appointments based on doctor availability.
* Enables online fee payments and prescription tracking for patients.
* Restricts system access to authorized personnel through secure login.
* Ensures efficient complaint handling by hospital authorities.
* Offers a user-friendly interface that speeds up hospital operations and improves overall patient experience.

The proposed system will significantly reduce manual workload, improve data security, and streamline hospital processes for both administrators and patients.

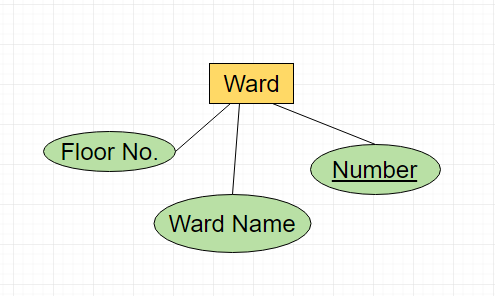
The database should be scalable to handle large number of users and growing amount of data. The database should be designed to ensure data reliability and database consistency.

The final solution should be presented in the form of an ER model and a database implementation using a RDBMS such as SQL.

Please remember, that this is not a final database system, it has the potential to grow into a vast database system. Thus, more and more features can easily be added to it, and it is not limited to its current capabilities. This database will currently take care of the basic requirement of a general hospital system.We will be able to keep records of wards, doctors, nurses, patients, diagnosis that patient receives, payment made by patients, appointments by patients and many others.

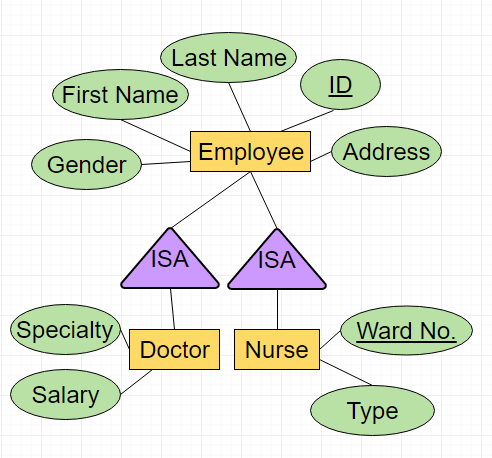
**Part 1. Entities in Hospital Domain Description**

* Ward
* Employee
  + Doctor
  + Nurse
* Research Lab
* Patient
* Admitted patient
* Appointment
* Diagnosis
* Payment



Ward Domain

* Entity: Ward
* Attributes:
  + Number (Primary key)
  + Floor No.
  + Ward name



Employee Domain

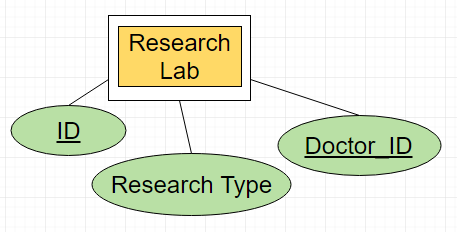
* Entity Set: Employees
* Entity: Employee
* Attributes:
  + ID (Primary key)
  + First Name
  + Last Name
  + Gender
  + Address

Doctor Domain

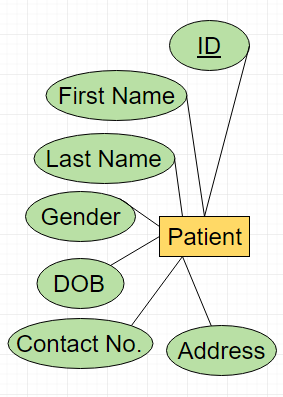
* Attributes: A Doctor has all employee attributes plus
  + Specialty
  + Salary

Nurse Domain

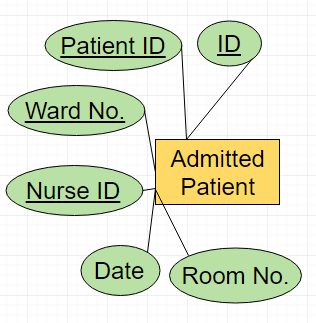
* Attributes: A Nurse has all employee attributes plus
  + Type
  + Ward No. (Foreign key)

Research Lab Domain

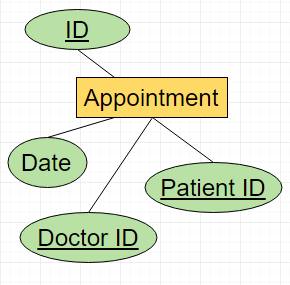
* Entity: Research lab
* Attributes:
  + ID (Primary key)
  + Research Type
  + Doctor\_ID (Foreign key)

Patient Domain

* Entity: Patient
* Attributes:
  + ID (Primary key)
  + First Name
  + Last Name
  + Gender
  + DOB
  + Contact No.
  + Address

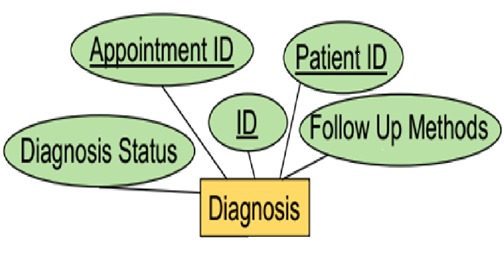
Admitted Patient Domain

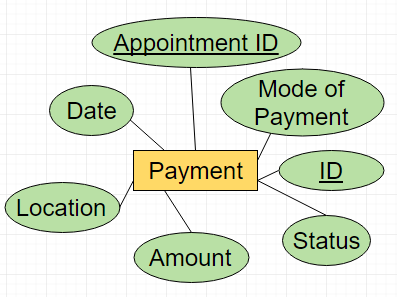
* Entity: Admitted Patient
* Attributes:
  + ID (Primary key)
  + Patient ID (Foreign Key)
  + Nurse ID (Foreign Key)
  + Ward No.(Foreign Key)
  + Date
  + Room No.

Appointment Domain

* Entity: Appointment
* Attributes:
  + ID (Primary key)
  + Patient ID (Foreign Key)
  + Doctor ID (Foreign Key)
  + Date

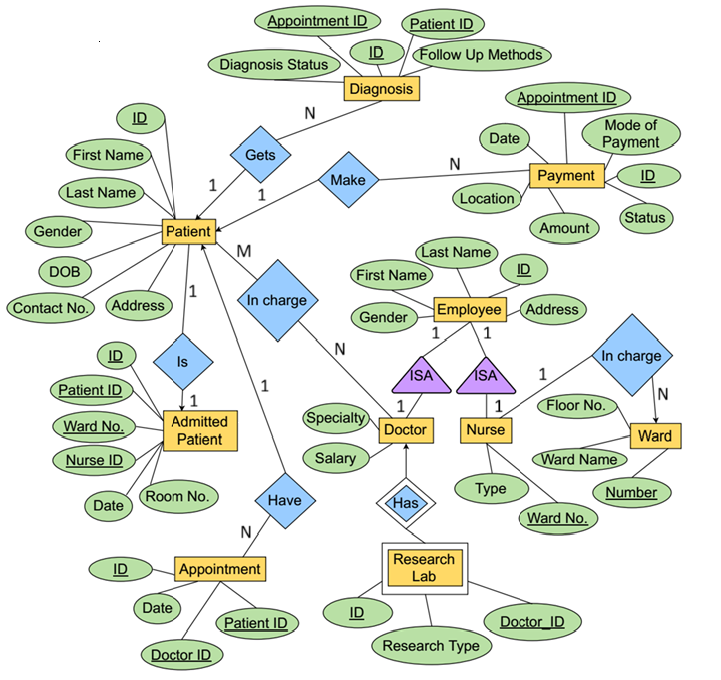
Diagnosis Domain

* Entity: Diagnosis
* Attributes: A Diagnosis has…
  + ID (Primary key)
  + Patient ID (Foreign Key)
  + Doctor ID (Foreign Key)
  + Date

Payment Domain

* Entity: Payment
* Attributes:
  + ID (Primary key)
  + Appointment ID (Foreign Key)
  + Mode of Payment
  + Status
  + Amount
  + Location
  + Date

**Part 2. ER DIAGRAM**



Relationships:

|  |  |  |  |
| --- | --- | --- | --- |
| **Domain 1** | **Domain 2** | **Relationship** | **Note** |
| Nurse | Ward | One-to-many | One ward can have multiple nurse, but one nurse can only belong to a single ward |
| Patient | Diagnosis | One-to-many | One patient can have multiple diagnosis |
| Patient | Payment | One-to-many | One patient can have multiple payments |
| Patient | Admitted Patient | One-to-one | One patient cannot have multiple admission at a time |
| Patient | Appointment | One-to-many | One patient can have multiple appointments |
| Doctor | Patient | Many-to-many | One patient can have multiple doctors, and one doctor can also have multiple patients |
| Doctor | Research Lab | Weak-Entity Set | Doctors may belong to a research lab |
|  |  |  |  |

**Part 3: Database Schema/Normal Forms**

Ward Domain will have the following SQL INSERT statement

CREATE TABLE Ward(

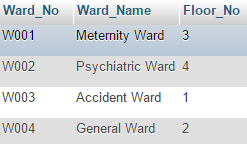
Ward\_No VARCHAR(8) NOT NULL PRIMARY KEY,

Ward\_Name VARCHAR(20) NOT NULL,

Floor\_No VARCHAR(1) NOT NULL

);

Sample looks like this:



Functional Dependencies = { Ward\_No🡪Ward\_Name,Floor\_No }

The schema is **free** from violations for:

– 3rd Normal form (all column depends on Ward\_No column)

– Boyce-Codd Normal Form (All function dependencies have a key on the left-hand side)

– 4th Normal Form (there are no multi value dependencies)

Doctor Domain will have the following SQL INSERT statement

CREATE TABLE Doctor(

Doctor\_ID VARCHAR(8) NOT NULL PRIMARY KEY,

Doctor\_First\_Name VARCHAR(20) NOT NULL,

Doctor\_Last\_Name VARCHAR(20) NOT NULL,

Doctor\_Gender VARCHAR(1) NOT NULL,

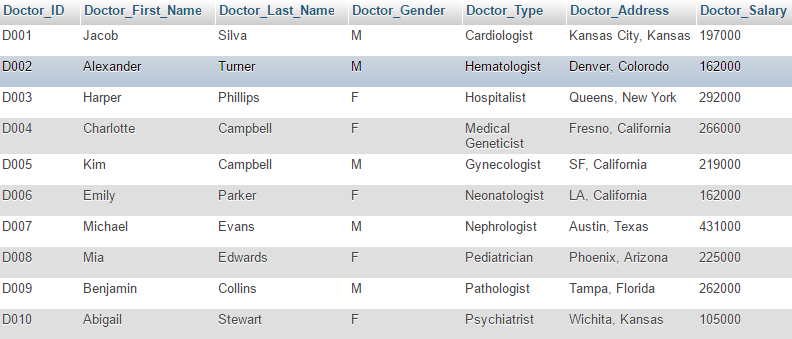
Doctor\_Specialty VARCHAR(20) NOT NULL,

Doctor\_Address VARCHAR(20) NOT NULL,

Doctor\_Salary VARCHAR(10) NOT NULL

);

Sample looks like this:



Functional Dependencies = { Doctor\_ID 🡪 Doctor\_first\_name,doctor\_last\_name,Doctor\_Gender,Doctor\_Type,Doctor\_Address,Doctor\_Salary }

The schema is **free** from violations for:

– 3rd Normal form (all column depends on Doctor\_ID column)

– Boyce-Codd Normal Form (All function dependencies have a key on the left-hand side)

– 4th Normal Form (there are no multi value dependencies)

Nurse Domain will have the following SQL INSERT statement

CREATE TABLE Nurse(

Nurse\_ID VARCHAR(8) NOT NULL PRIMARY KEY,

Nurse\_First\_Name VARCHAR(20) NOT NULL,

Nurse\_Last\_Name VARCHAR(20) NOT NULL,

Nurse\_Gender VARCHAR(1) NOT NULL,

Ward\_No VARCHAR(8) NOT NULL,

Nurse\_Type Varchar(20),

FOREIGN KEY (Ward\_No) REFERENCES Ward(Ward\_No)

)

Sample looks like this:



Functional Dependencies = {Nurse\_id🡪 Nurse\_first\_name,Nurse\_Last\_name,Nurse\_Gender,Ward\_No,Nurse\_Type}

The schema is **free** from violations for:

– 3rd Normal form (all column depends on Doctor\_ID column)

– Boyce-Codd Normal Form (All function dependencies have a key on the left-hand side)

The schema is a violation for:

– 4th Normal Form (Column Nurse\_Type has multi value dependencies)

Patient Domain will have the following SQL INSERT statement

CREATE TABLE Patient(

Patient\_ID VARCHAR(8) NOT NULL PRIMARY KEY,

Patient\_First\_Name VARCHAR(20) NOT NULL,

Patient\_Last\_Name VARCHAR(20) NOT NULL,

Patient\_Gender VARCHAR(1) NOT NULL,

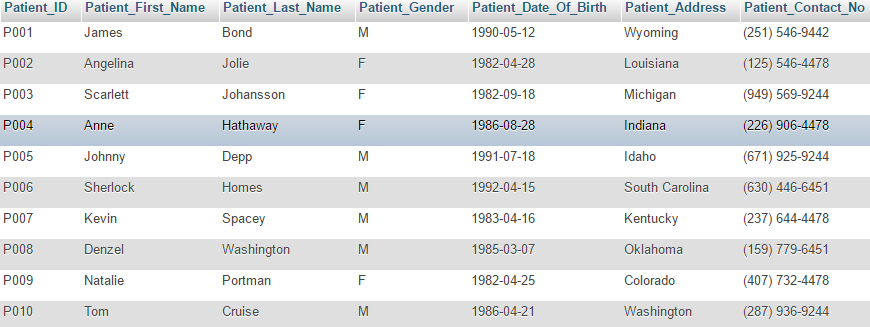
Patient\_Date\_Of\_Birth DATE NOT NULL,

Patient\_Address VARCHAR(20) NOT NULL,

Patient\_Contact\_No VARCHAR(24) NOT NULL

);

Sample looks like this:



Functional Dependencies = {Patient\_ID 🡪 Patient\_first\_name,Patient\_last\_name,patient\_gender,patient\_dob,Patient\_Address,Patient\_Contact\_NO}

The schema is **free** from violations for:

– 3rd Normal form (all column depends on Patient\_ID column)

– Boyce-Codd Normal Form (All function dependencies have a key on the left-hand side)

– 4th Normal Form (there are no multi value dependencies)

Admitted Patient Domain will have the following SQL INSERT statement

CREATE TABLE Admitted\_Patient(

Admitted\_ID VARCHAR(8) NOT NULL PRIMARY KEY,

Patient\_ID VARCHAR(8) NOT NULL,

Ward\_No VARCHAR(8) NOT NULL,

Nurse\_ID VARCHAR(8) NOT NULL,

Admitted\_Date\_Time DATETIME NOT NULL,

Room\_No VARCHAR(20) NOT NULL,

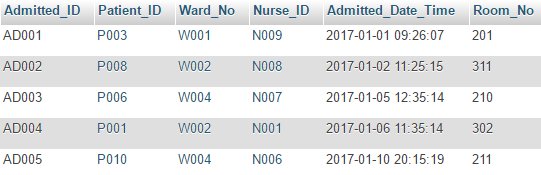
FOREIGN KEY (Patient\_ID) REFERENCES Patient(Patient\_ID),

FOREIGN KEY (Ward\_No) REFERENCES Ward(Ward\_No),

FOREIGN KEY (Nurse\_ID) REFERENCES Nurse(Nurse\_ID)

)

Sample looks like this:



Functional Dependencies = {Admitted\_id🡪 Patient\_id,Ward\_No,Nurse\_id,Admitted\_date\_time,Room\_no}

The schema is **free** from violations for:

– 3rd Normal form (all column depends on Admitted\_ID column)

– Boyce-Codd Normal Form (All function dependencies have a key on the left-hand side)

– 4th Normal Form (there are no multi value dependencies)

Appointment Domain will have the following SQL INSERT statement

CREATE TABLE Appointment(

Appointment\_ID VARCHAR(8) NOT NULL PRIMARY KEY,

Patient\_ID VARCHAR(8) NOT NULL,

Doctor\_ID VARCHAR(8) NOT NULL,

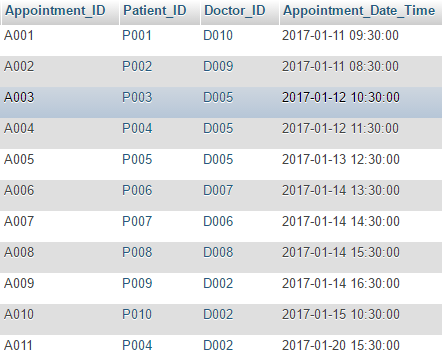
Appointment\_Date\_Time DATETIME NOT NULL,

FOREIGN KEY (Doctor\_ID) REFERENCES Doctor(Doctor\_ID),

FOREIGN KEY (Patient\_ID) REFERENCES Patient(Patient\_ID)

);

Sample looks like this:



Functional\_Dependencies = {Appointment\_id --> Patient\_id,Doctor\_id,Appointment\_date\_time}

The schema is **free** from violations for:

– 3rd Normal form (all column depends on Appointment\_ID column)

– Boyce-Codd Normal Form (All function dependencies have a key on the left-hand side)

– 4th Normal Form (there are no multi value dependencies)

Diagnosis Domain will have the following SQL INSERT statement

CREATE TABLE Diagnosis(

Diagnosis\_ID VARCHAR(8) NOT NULL PRIMARY KEY,

Appointment\_ID VARCHAR(8) NOT NULL,

Patient\_ID VARCHAR(8) NOT NULL,

Diagnosis\_Status VARCHAR(50) NOT NULL,

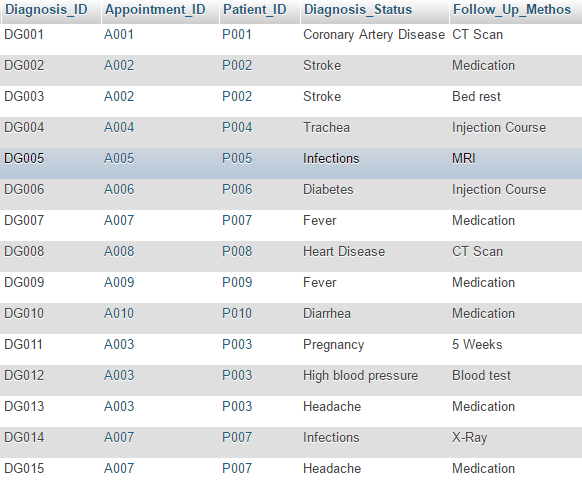
Follow\_Up\_Methods VARCHAR(50) NOT NULL,

FOREIGN KEY (Appointment\_ID) REFERENCES Appointment(Appointment\_ID),

FOREIGN KEY (Patient\_ID) REFERENCES Patient(Patient\_ID)

);

Sample looks like this:



Functional Dependencies= {Diagnosis\_id 🡪 Appointment\_id,Patient\_id,Diagnosis\_status,Follow\_up\_methods}

The schema is **free** from violations for:

– 3rd Normal form (all column depends on Diagnosis\_ID column)

– Boyce-Codd Normal Form (All function dependencies have a key on the left-hand side)

– 4th Normal Form (there are no multi value dependencies)

NOTE: Though Diagnosis\_Status column and Follow\_Up\_Methodes column have some data that are common, but these fields are text (comment) fields, they can contain anything, nothing (null value) or duplicates.

Payment Domain will have the following SQL INSERT statement

CREATE TABLE Payment(

Payment\_ID VARCHAR(8) NOT NULL PRIMARY KEY,

Appointment\_ID VARCHAR(8) NOT NULL,

Mode\_Of\_Payment VARCHAR(20) NOT NULL,

Payment\_Date\_Time DATETIME NOT NULL,

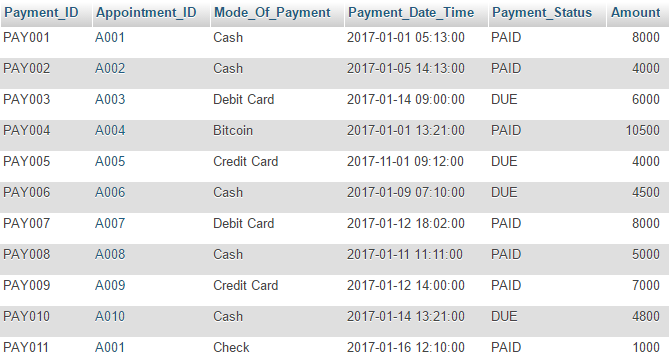
Payment\_Status VARCHAR(20) NOT NULL,

Payment\_Location VARCHAR(20) NOT NULL,

Amount INT NOT NULL,

FOREIGN KEY (Appoinment\_ID) REFERENCES Appointment(Appointment\_ID)

);



Functional dependencies = {Payment\_id🡪 Appointment\_id,Mode\_of\_payment,payment\_date\_time,Payment\_status,Amount}

The schema is **free** from violations for:

– 3rd Normal form (all column depends on Payment\_ID column)

– Boyce-Codd Normal Form (All function dependencies have a key on the left-hand side)

The schema is a violation for:

– 4th Normal Form (Column Mode\_Of\_Payment and Payment\_Status has multi value dependencies)

Research Lab Domain will have the following SQL INSERT statement

CREATE TABLE Research\_Lab(

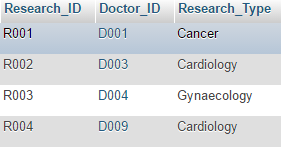
Research\_ID VARCHAR(8) NOT NULL PRIMARY KEY,

Doctor\_ID VARCHAR(8) NOT NULL,

Research\_Type VARCHAR(20) NOT NULL,

FOREIGN KEY (Doctor\_ID) REFERENCES Doctor(Doctor\_ID)

);



Functional Dependencies = {Research\_id 🡪 Doctor\_ID , Research\_type}

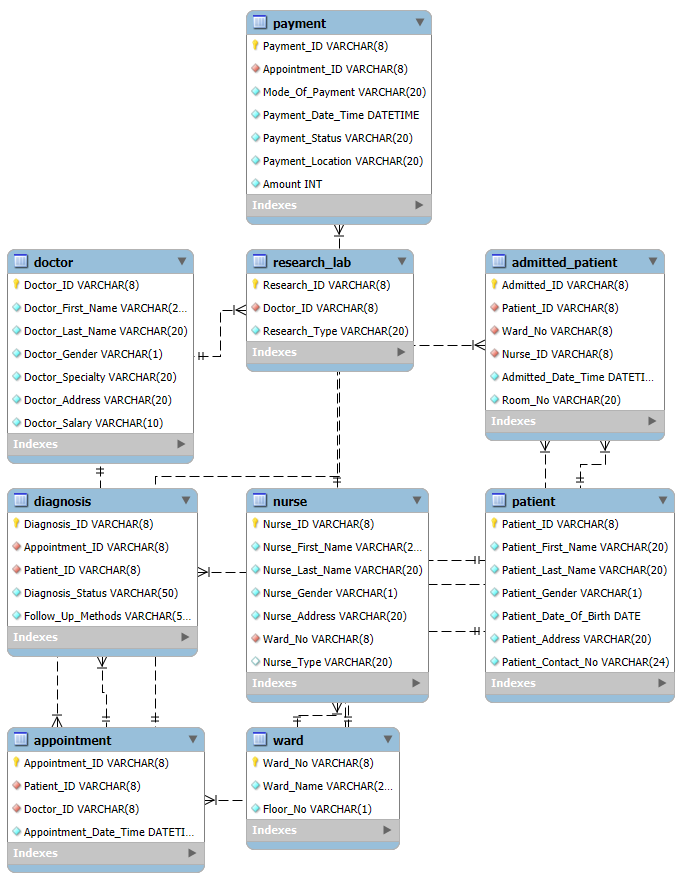
The schema is **free** from violations for:

– 3rd Normal form (all column depends on Research\_ID column)

– Boyce-Codd Normal Form (All function dependencies have a key on the left-hand side)

– 4th Normal Form (there are no multi value dependencies)

**Part-4 : Relational Schema**



**Part 5: SQL Queries to Insert Data**

**1.Ward**

INSERT INTO Ward VALUES

('W003','Accident Ward','1'),

('W001','Meternity Ward','2'),

('W004','General Ward','2'),

('W002','Psychiatric Ward','3');

2.Doctor

INSERT INTO Doctor Values

('D001','Jacob','Silva','M','Cardiologist','Kansas City, Kansas','197000'),

('D002','Alexander','Turner','M','Hematologist','Denver, Colorodo','162000'),

('D003','Harper','Phillips ','F','Hospitalist','Queens, New York','292000'),

('D004','Charlotte','Campbell','F','Medical Geneticist','Fresno, California','266000'),

('D005','Kim','Campbell','M','Gynecologist','SF, California','219000'),

('D006','Emily','Parker','F','Neonatologist','LA, California','162000'),

('D007','Michael','Evans','M','Nephrologist','Austin, Texas','431000'),

('D008','Mia','Edwards','F','Pediatrician','Phoenix, Arizona','225000'),

('D009','Benjamin','Collins','M','Pathologist','Tampa, Florida','262000'),

('D010','Abigail','Stewart','F','Psychiatrist','Wichita, Kansas','105000');

3.Nurse

INSERT INTO Nurse VALUES

('N001','Noah', 'Smith','M','W001','Head','Kansas'),

('N002','Emma', 'Williams','F','W001','General','Colorodo'),

('N003','Liam', 'Johnson','M','W002','Head','Colorodo'),

('N004','Olivia', 'Jones','F','W003','Head','Colorodo'),

('N005','Jacob', 'Brown','M','W004','Head','Colorodo'),

('N006','Ava', 'Miller','F','W002','General','Kansas'),

('N007','Ethan', 'Moore','M','W003','General','Oaklohama'),

('N008','Isabella', 'Thomas','F','W004','General','Kansas'),

4.Patient

INSERT INTO Patient VALUES

('P001','James','Bond','M','1990-05-12','Alabama','(251) 546-9442'),

('P002','Angelina','Jolie','F','1982-04-28','Louisiana','(125) 546-4478'),

('P003','Scarlett','Johansson','F','1982-09-18','Michigan','(949) 569-4371'),

('P004','Anne','Hathaway','F','1986-08-28','Indiana','(226) 906-2721'),

('P005','Johnny','Depp','M','1991-07-18','Idaho','(671) 925-1352'),

('P006','Sherlock','Homes','M','1992-04-15','South Carolina','(630) 446-8851'),

('P007','Kevin','Spacey','M','1983-04-16','Kentucky','(237) 644-9244'),

('P008','Denzel','Washington','M','1985-03-07','Oklahoma','(159) 779-4578'),

('P009','Natalie','Portman','F','1982-04-25','Colorado','(407) 732-7516'),

('P010','Tom','Cruise','M','1986-04-21','Washington','(287) 936-6451');

5.Admitted\_Patient

INSERT INTO Admitted\_Patient VALUES

('AD001','P003','W001','N009','2025-01-01 09:26:07','201'),

('AD002','P008','W002','N008','2025-01-02 11:25:15','311'),

('AD003','P006','W004','N007','2025-01-05 12:35:14','210'),

('AD004','P001','W002','N001','2025-01-06 11:35:14','302'),

('AD005','P010','W004','N006','2025-01-10 20:15:19','211');

6.Appointment

INSERT INTO Appointment VALUES

('A001','P001','D010','2025-01-11 09:30:00'),

('A002','P002','D009','2025-01-11 08:30:00'),

('A003','P003','D005','2025-01-12 10:30:00'),

('A004','P004','D005','2025-01-12 11:30:00'),

('A005','P005','D005','2025-01-13 12:30:00'),

('A006','P006','D007','2025-01-14 13:30:00');

7.Diagnosis

INSERT INTO Diagnosis VALUES

('DG001','A001','P001','Coronary Artery Disease','CT Scan'),

('DG002','A002','P002','Stroke','Medication'),

('DG003','A002','P002','Stroke','Bed rest'),

('DG004','A004','P004','Trachea','Injection Course'),

('DG005','A005','P005','Infections','MRI'),

('DG006','A006','P006','Diabetes','Injection Course'),

('DG007','A007','P007','Fever','Medication'),

('DG008','A008','P008','Heart Disease','CT Scan'),

('DG009','A009','P009','Fever','Medication'),

('DG010','A010','P010','Diarrhea','Medication'),

('DG011','A003','P003','Pregnancy','5 Weeks'),

('DG012','A003','P003','High blood pressure','Blood test'),

('DG013','A003','P003','Headache','Medication'),

('DG014','A007','P007','Infections','X-Ray'),

('DG015','A007','P007','Headache','Medication');

8.Payment

INSERT INTO Payment VALUES

('PAY001','A001','Cash','2025-01-01 05:13:00','PAID','8000'),

('PAY002','A002','Cash','2025-01-05 14:13:00','PAID','4000'),

('PAY003','A003','Debit Card','2025-01-14 09:00:00','DUE','6000'),

('PAY004','A004','Bitcoin','2025-01-01 13:21:00','PAID','10500'),

('PAY005','A005','Credit Card','2025-11-01 09:12:00','DUE','4000'),

('PAY006','A006','Cash','2025-01-09 07:10:00','DUE','4500'),

('PAY007','A007','Debit Card','2025-01-12 18:02:00','PAID','8000'),

('PAY008','A008','Cash','2025-01-11 11:11:00','PAID','5000'),

('PAY009','A009','Credit Card','2025-01-12 14:00:00','PAID','7000'),

('PAY010','A010','Cash','2025-01-14 13:21:00','DUE','4800'),

('PAY011','A001','Check','2025-01-16 12:10:00','PAID','1000');

9.Research Lab

INSERT INTO Research\_Lab VALUES

('R001','D001','Cancer'),

('R002','D003','Cardiology'),

('R003','D004','Gynaecology'),

('R004','D009','Cardiology');

**Part 6: Database in Action**

1.  **To find a patient by address**

SELECT \*

FROM Patient

WHERE Patient\_Address='South Carolina';

Result:



2. **To find a specific doctor**

SELECT \*

FROM Doctor

WHERE Doctor\_Gender='M' AND Doctor\_Last\_Name='Campbell';

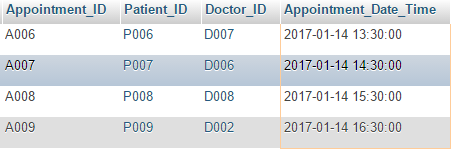


3. **To find appointment on a given date**

SELECT \*

FROM Appointment

WHERE DATEDIFF(Appointment\_Date\_Time, '2025-01-14')=0



4. **To find the phone number of female patients who are suffering from strokes**

SELECT Patient\_Contact\_No

FROM Patient

WHERE Patient\_ID IN

( SELECT Patient\_ID

FROM Diagnosis

WHERE Diagnosis\_Status ='Stroke')

)

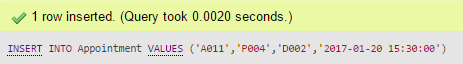
AND Patient\_Gender='F';



5. **To create a new appointment**

INSERT INTO Appointment VALUES

('A011','P004','D002','2025-01-20 15:30:00');

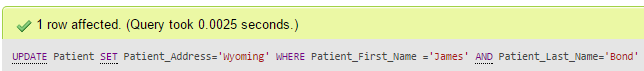


6. **To update the address of a given patient**

UPDATE Patient

SET Patient\_Address='Wyoming'

WHERE Patient\_First\_Name ='James' AND Patient\_Last\_Name='Bond';



7. **To find assigned doctors for a selected patient, and his salary**

SELECT Doctor\_First\_Name, Doctor\_Last\_Name, Doctor\_Salary

FROM Appointment a, Doctor d

WHERE

d.Doctor\_ID = a.Doctor\_ID

AND

a.Patient\_ID =

(

SELECT Patient\_ID FROM Patient

WHERE Patient\_First\_Name= 'Scarlett'

AND Patient\_Last\_Name= 'Johansson'

)

;



8. **To find number of upcoming appointments between dates**

SELECT count(Appointment\_ID) AS Appointments\_Upcomming

FROM Appointment

WHERE Appointment\_Date\_Time

BETWEEN '2025-01-14 00:00:00'

AND '2025-01-15 23:59:00';



9. **To find total amount paid by a single patient**

SELECT SUM(Amount) AS Total\_Amount\_Paid\_By\_James\_Bond

FROM Payment p, Appointment a

WHERE

a.Appointment\_ID = p.Appointment\_ID

AND

a.Patient\_ID = (

SELECT Patient\_ID FROM Patient

WHERE Patient\_First\_Name = 'James'

AND Patient\_Last\_Name = 'Bond'

);



10. **To find patient with appointment that listed ‘fever’ as their illness**

SELECT Patient\_First\_Name, Patient\_Last\_Name

FROM Patient

WHERE Patient\_ID IN

(SELECT Patient\_ID FROM Appointment WHERE Appointment\_ID IN

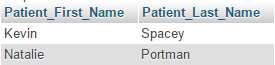
(SELECT Appointment\_ID

FROM Diagnosis

WHERE Diagnosis\_Status='Fever')

)

ORDER BY Patient\_ID;

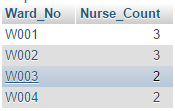


11. **To find number of nurse assigned for each ward**

SELECT Ward\_No, COUNT(\*) AS Nurse\_Count

FROM Nurse

GROUP BY Ward\_No



12. **To find nurses in a certain ward with patients in it**

SELECT Nurse\_First\_Name, Nurse\_Last\_Name

FROM Nurse

WHERE Nurse\_ID IN (

SELECT Nurse\_ID FROM Admitted\_Patient

WHERE Ward\_No='W001'

);



13. **To find maximum payment received by the hospital by a single patient**

SELECT Mode\_Of\_Payment, Amount AS Max\_Payment\_Made\_By\_Patient

FROM Payment

WHERE Amount=(SELECT MAX(Amount) FROM Payment);



14. **To count the number of male patient that are currently admitted in the hospital**

SELECT COUNT(Admitted\_ID) AS Male\_Patient\_COUNT

FROM Admitted\_Patient

WHERE Patient\_ID IN(

SELECT Patient\_ID FROM Patient

WHERE Patient\_Gender='M'

);

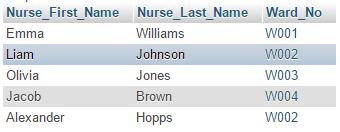


15. **To Find nurses who are not allotted to any ward**

SELECT Nurse\_First\_Name, Nurse\_Last\_Name, Ward\_No

FROM Nurse

WHERE Nurse\_ID NOT IN (SELECT Nurse\_ID FROM Admitted\_Patient);



16**. To Find doctors who are treating ‘Diabetes’**

SELECT DISTINCT doc.Doctor\_ID, Doctor\_First\_Name, Doctor\_Last\_Name

FROM Doctor doc, Appointment a, Diagnosis dig

WHERE doc.Doctor\_ID = a.Doctor\_ID

AND dig.Diagnosis\_Status = 'Diabetes'

AND dig.Appointment\_ID = a.Appointment\_ID

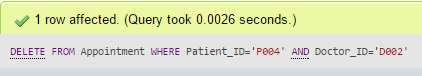
ORDER BY Doctor\_ID;



17. **To cancel an appointment**

DELETE FROM Appointment

WHERE Patient\_ID='P004' AND Doctor\_ID='D002';



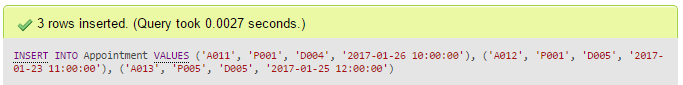
18. **To create multiple appointments at the same time**

INSERT INTO Appointment VALUES

('A011', 'P001', 'D004', '2025-01-26 10:00:00'),

('A012', 'P001', 'D005', '2025-01-23 11:00:00'),

('A013', 'P005', 'D005', '2025-01-25 12:00:00');

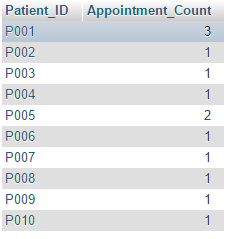


19. To verify **[Appointment count for Patient P001 is now 3]**

SELECT Patient\_ID, COUNT(\*) AS Appointment\_Count

FROM Appointment

GROUP BY Patient\_ID;



20.To **find Doctors who have multiple appointments**

SELECT Doctor\_ID,COUNT(Appointment\_ID) AS Appointment\_Count

FROM Appointment

GROUP BY Doctor\_ID

HAVING COUNT(Appointment\_ID) >= 2;

