

# NETAJI SUBHAS UNIVERSITY OF TECHNOLOGY



## Database Management Systems ICCS09

### Hospital Management System (HMS)

#### **Submitted to**

Mrs. Sushma Nagpal

#### **Submitted by**

Krish Gupta      2022UIC3507

Devansh Behl      2022UIC3582

Meet Singh      2022UIC8057

## **Overview**

The Hospital Management System (HMS) is a comprehensive solution designed to streamline and digitalize various processes within a hospital environment. This project aims to enhance the efficiency, accuracy, and accessibility of information across different departments.

## **Problem Statement**

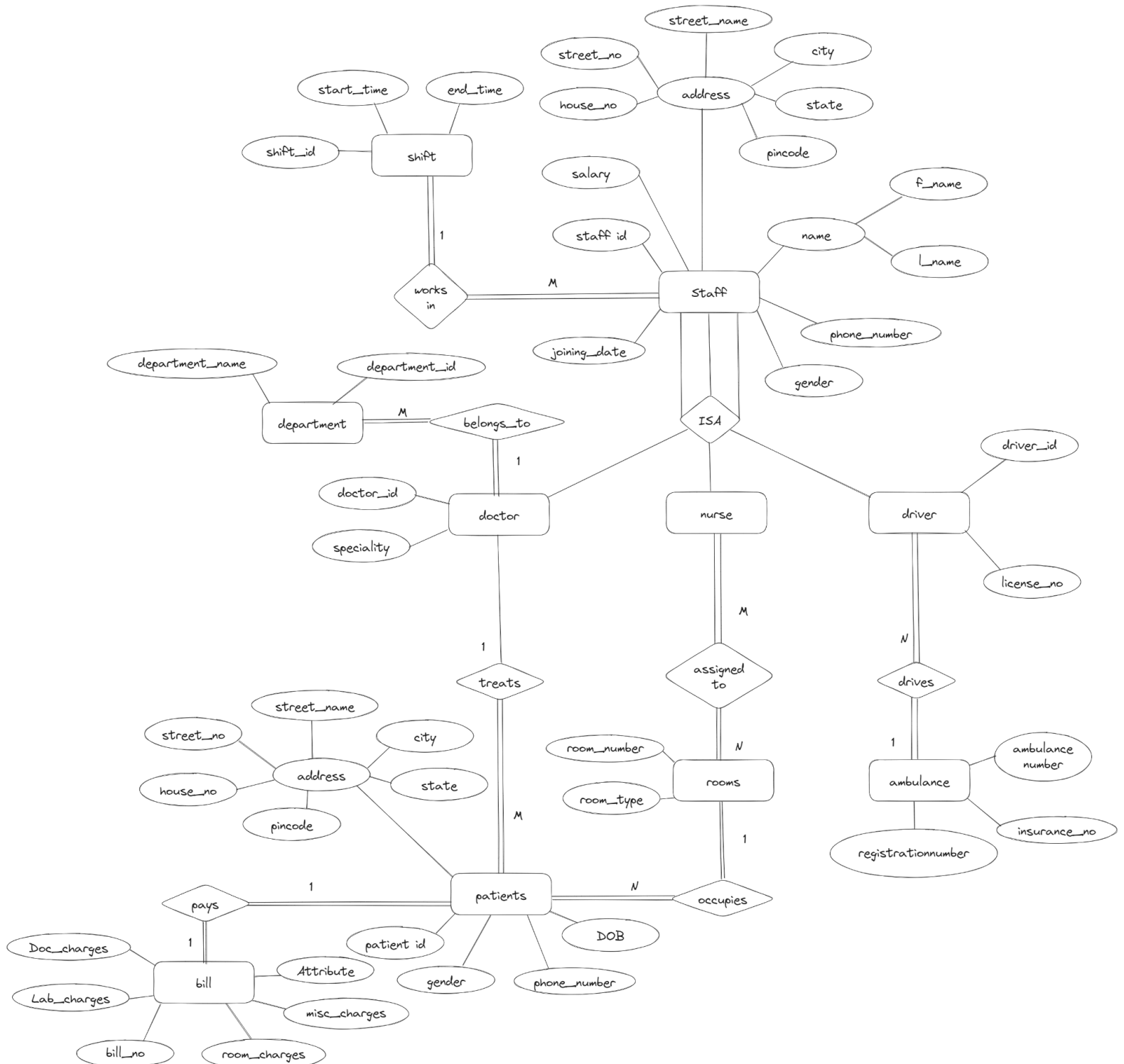
Healthcare institutions often face challenges in managing patient information, appointments, billing, and other administrative tasks. The Hospital Management System is developed to address these challenges by introducing a centralized digital platform.

The current hospital management system is plagued by inefficiencies and limitations that hinder the delivery of optimal healthcare services. These issues range from administrative bottlenecks to patient care challenges, and they impede the overall effectiveness of the healthcare institution. As such, there is an urgent need to develop and implement an advanced hospital management system.

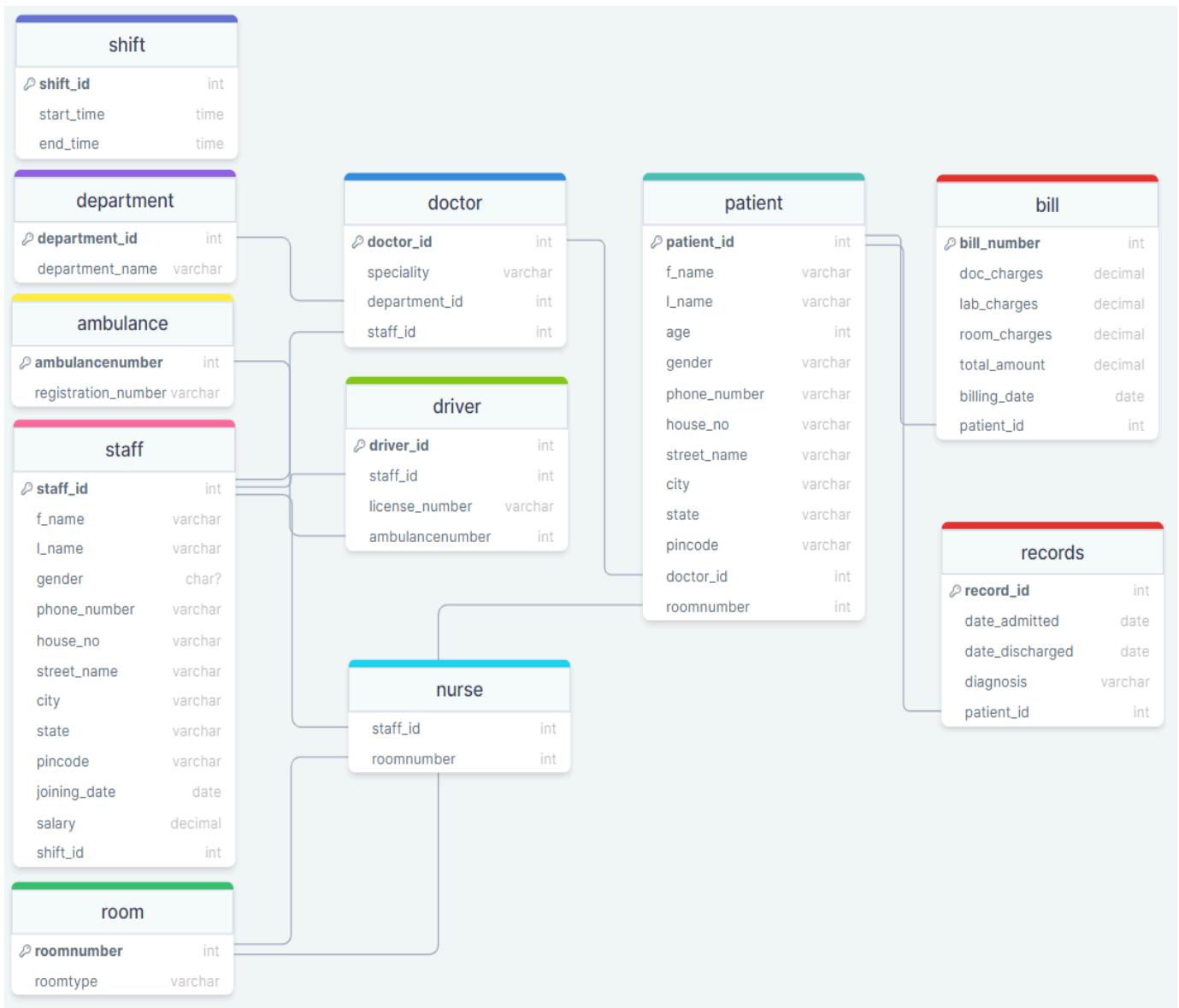
## **Purpose of Project**

- Automate hospital processes for improved efficiency.
- Enhance patient care by facilitating quick access to medical records.
- Improve resource management, including staff scheduling and inventory control.
- Ensure accurate billing and financial tracking.

## ER Diagram

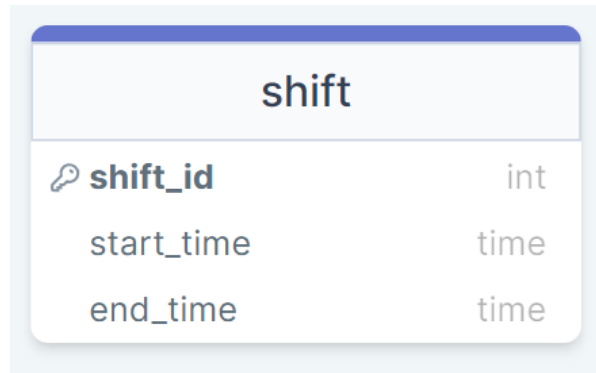


## Relational Schema



## Function Dependencies and Third Normal Form (3NF)

### 1. Shift



shift	
🔑 shift_id	int
start_time	time
end_time	time

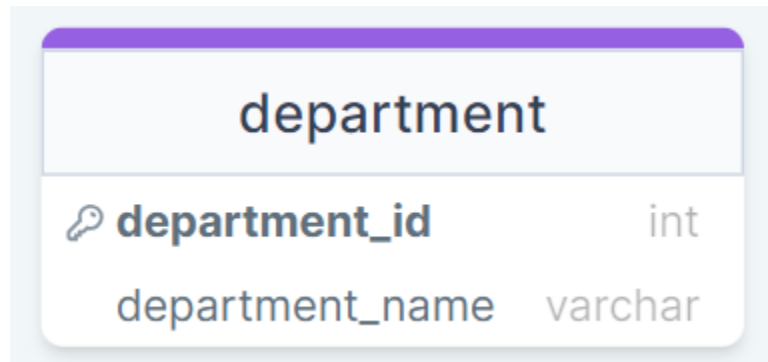
- **Functional Dependencies**

- $\text{shift\_id} \rightarrow \text{start\_time}$
- $\text{shift\_id} \rightarrow \text{end\_time}$

- **Normal Form**

- 1st Normal Form (1NF): The table already meets 1NF since there are no repeating groups, and each cell contains atomic values.
- 2nd Normal Form (2NF): Since there's only one candidate key (`shift_id`), and both `start_time` and `end_time` are fully functionally dependent on the entire candidate key, we don't have partial dependencies. Thus, the table is already in 2NF.
- 3rd Normal Form (3NF): In this step, we check for transitive dependencies. Since there's only one non-prime attribute, `start_time` and `end_time` are both attributes that directly depend on the primary key. There are no transitive dependencies, so the table is already in 3NF.

## 2. Department



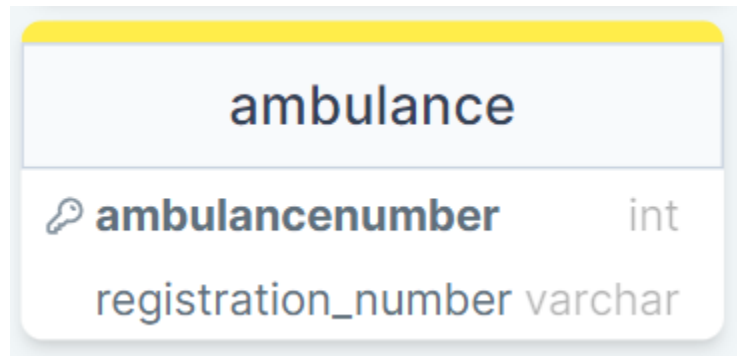
- **Functional Dependencies**

- $\text{department\_id} \rightarrow \text{department\_name}$

- **Normal Form**

- 1st Normal Form (1NF): The table already meets 1NF since there are no repeating groups, and each cell contains atomic values.
- 2nd Normal Form (2NF): Since there's only one candidate key (`department_id`), and `department_name` is fully functionally dependent on the entire candidate key, we don't have partial dependencies. Thus, the table is already in 2NF.
- 3rd Normal Form (3NF): In this step, we need to ensure that there are no transitive dependencies. Since there's only one non-prime attribute (`department_name`), and it is directly dependent on the primary key, there are no transitive dependencies.

### 3. Ambulance



- **Functional Dependencies**

1. ambulance\_number  $\rightarrow$  registration\_number

- **Normal Form**

- First Normal Form (1NF): The table seems to be in 1NF because each attribute contains atomic values, and there are no repeating groups.
- Second Normal Form (2NF): Since there is only one non-prime attribute (registration\_number), there are no partial dependencies. Thus, the table already satisfies the requirements of 2NF.
- Third Normal Form (3NF): To achieve 3NF, we need to ensure that there are no transitive dependencies. In this schema, there is only one non-prime attribute, so there are no transitive dependencies to address.

## 4. Staff

staff	
 <b>staff_id</b>	int
f_name	varchar
l_name	varchar
gender	char?
phone_number	varchar
house_no	varchar
street_name	varchar
city	varchar
state	varchar
pincode	varchar
joining_date	date
salary	decimal
shift_id	int

- **Functional Dependencies**

- $\text{staff\_id} \rightarrow \text{speciality}$ : Each doctor's  $\text{staff\_id}$  uniquely determines their specialty.
- $\text{staff\_id} \rightarrow \text{department\_id}$ : Each doctor's  $\text{staff\_id}$  uniquely determines department.
- $\text{staff\_id} \rightarrow \text{license\_number}$ : Each driver's  $\text{staff\_id}$  uniquely determines their license number.
- $\text{staff\_id} \rightarrow \text{ambulance\_number}$ : Similarly, each driver's  $\text{staff\_id}$  uniquely determines the ambulance they are assigned to.
- $\text{staff\_id} \rightarrow \text{room\_number}$ : Each nurse's  $\text{staff\_id}$  uniquely determines the room number they are assigned to.



- **Normal Form**

- First Normal Form (1NF):The table already seems to be in 1NF because each attribute contains atomic values, and there are no repeating groups.
- Second Normal Form (2NF):The table is already in 2NF because there are no partial dependencies. Each non-prime attribute depends on the entire primary key.
- Third Normal Form (3NF):To achieve 3NF, we need to ensure there are no transitive dependencies. There is a transitive dependency between staff and shift\_id. Since the shift details might not be directly related to the staff member, we should move this attribute to another table.

**Normalized Schema:**

staff\_info:

staff(staff\_id (primary key), f\_name, l\_name, gender, phone\_number, house\_no, street\_name, city, state, pincode, joining\_date, salary)

shift\_assignment:

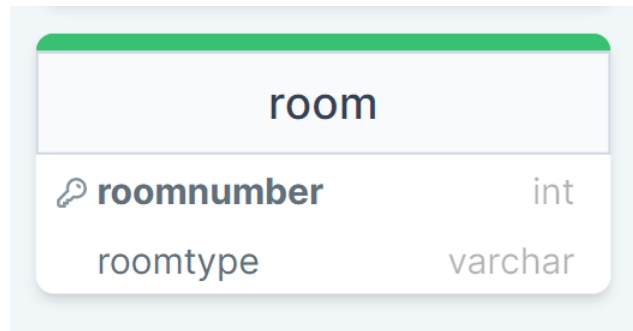
shift\_assignment(staff\_id (foreign key), shift\_id (foreign key), primary key(staff\_id, shift\_id))


1. doctor(doctor\_id(primary key), speciality, department\_id, staff\_id(foreign key)):  
- No changes required as it's a separate entity.

2. driver(driver\_id (primary key), staff\_id(foreign key), license\_number, ambulance\_number):  
- No changes required as it's a separate entity.

3. nurse(staff\_id (foreign key), room\_number):  
- No changes required as it's a separate entity.

## 5. Room



room	
 <b>roomnumber</b>	int
roomtype	varchar


- **Functional Dependencies**

- roomnumber  $\rightarrow$  roomtype
- roomnumber  $\rightarrow$  staff\_id
- roomnumber  $\rightarrow$  patient\_id

- **Normal Form**

- First Normal Form (1NF): The table appears to be in 1NF because each attribute contains atomic values, and there are no repeating groups.
- Second Normal Form (2NF): The table is already in 2NF because there are no partial dependencies. Each non-prime attribute depends on the entire primary key.
- Third Normal Form (3NF): To achieve 3NF, we need to ensure there are no transitive dependencies..Since roomnumber is a foreign key in both the "nurse" and "patient" tables, and it directly relates to the room type, there are no transitive dependencies to address.

## 6. Doctor

doctor	
 <b>doctor_id</b>	int
speciality	varchar
department_id	int
staff_id	int


- **Functional Dependencies**

- $\text{doctor\_id} \rightarrow \text{speciality}$
- $\text{doctor\_id} \rightarrow \text{staff\_id}$

- **Normal Form**

- First Normal Form (1NF): The table appears to be in 1NF because each attribute contains atomic values, and there are no repeating groups.
- Second Normal Form (2NF): The table is already in 2NF because there are no partial dependencies. Each non-prime attribute depends on the entire primary key.
- Third Normal Form (3NF): To achieve 3NF, we need to ensure there are no transitive dependencies. Since `doctor_id` is a foreign key in the "patient" table and it directly relates to the doctor's specialty and staff ID, there are no transitive dependencies to address.

## 7. Driver

driver	
 <b>driver_id</b>	int
staff_id	int
license_number	varchar
ambulancenum	int

- **Functional Dependencies**

- $\text{driver\_id} \rightarrow \text{staff\_id}$
- $\text{driver\_id} \rightarrow \text{license\_number}$
- $\text{driver\_id} \rightarrow \text{ambulancenum}$

- **Normal Form**

- First Normal Form (1NF): The table already appears to be in 1NF because each attribute contains atomic values, and there are no repeating groups.
- Second Normal Form (2NF): The table is already in 2NF because there are no partial dependencies. Each non-prime attribute depends on the entire primary key.
- Third Normal Form (3NF): To achieve 3NF, we need to ensure there are no transitive dependencies. There are no transitive dependencies in the original schema.

## 8. Driver

nurse	
staff_id	int
roomnumber	int

- **Functional Dependencies**

- $(\text{staff\_id}, \text{roomnumber}) \rightarrow \text{None}$

- **Normal Form**

- First Normal Form (1NF): The table is already in 1NF as there are no repeating groups, and all attributes are atomic.
- Second Normal Form (2NF): Since there's no composite primary key, and both `staff_id` and `roomnumber` are attributes directly dependent on the entire primary key, the table is already in 2NF.
- Third Normal Form (3NF): There are no attributes that depend on any non-primary key attributes, so there are no transitive dependencies. Therefore, the table is already in 3NF.

## 9. Patient

patient	
 <b>patient_id</b>	int
f_name	varchar
l_name	varchar
age	int
gender	varchar
phone_number	varchar
house_no	varchar
street_name	varchar
city	varchar
state	varchar
pincode	varchar
doctor_id	int
roomnumber	int


- **Functional Dependencies**

- $\text{patient\_id} \rightarrow \text{f\_name}, \text{l\_name}, \text{age}, \text{gender}, \text{phone\_number}, \text{house\_no}, \text{street\_name}, \text{city}, \text{state}, \text{pincode}, \text{doctor\_id}, \text{roomnumber}$

- **Normal Form**

- First Normal Form (1NF): The table is already in 1NF as there are no repeating groups, and all attributes are atomic.
- Second Normal Form (2NF): Since there's no composite primary key, and all non-prime attributes (`f_name`, `l_name`, `age`, `gender`, `phone_number`, `house_no`, `street_name`, `city`, `state`, `pincode`, `doctor_id`, `roomnumber`) are fully functionally dependent on the entire primary key (`patient_id`), the table is already in 2NF.
- Third Normal Form (3NF): There are no attributes that depend on any non-primary key attributes, so there are no transitive dependencies. Therefore, the table is already in 3NF.

## 10. Bill

bill	
 <b>bill_number</b>	int
doc_charges	decimal
lab_charges	decimal
room_charges	decimal
total_amount	decimal
billing_date	date
patient_id	int

- **Functional Dependencies**

- $\text{bill\_number} \rightarrow \text{doc\_charges}, \text{lab\_charges}, \text{room\_charges}, \text{total\_amount}, \text{billing\_date}, \text{patient\_id}$

- **Normal Form**

- First Normal Form (1NF): Already satisfied.
- Second Normal Form (2NF): Since there's only one candidate key (**bill\_number**), and all other attributes (**doc\_charges, lab\_charges, room\_charges, total\_amount, billing\_date, patient\_id**) are fully functionally dependent on the entire candidate key, we don't have partial dependencies. Thus, the table is already in 2NF.
- Third Normal Form (3NF): In this step, we need to ensure that there are no transitive dependencies. Since there are no attributes that depend on any non-primary key attributes, there are no transitive dependencies.

## 11. Records

records	
🔑 record_id	int
date_admitted	date
date_discharged	date
diagnosis	varchar
patient_id	int

- **Functional Dependencies**

- record\_id → date\_admitted, date\_discharged, diagnosis, patient\_id
- patient\_id → None

- **Normal Form**

- First Normal Form (1NF): Already satisfied.
- Second Normal Form (2NF): Since there's only one candidate key (record\_id), and all other attributes (date\_admitted, date\_discharged, diagnosis, patient\_id) are fully functionally dependent on the entire candidate key, we don't have partial dependencies. Thus, the table is already in 2NF.
- Third Normal Form (3NF): In this step, we need to ensure that there are no transitive dependencies. Since there are no attributes that depend on any non-primary key attributes, there are no transitive dependencies.