

# Hospital

# Patient Manager



# Introduction

**Member 1**

Khuất Đình Trung

ID : 11247362

**Member 2**

Lê Sỹ Huy

ID : 11247296

**Member 3**

Bé Thành Đạt

ID : 11247272

# Chapter 1:

# Introduction



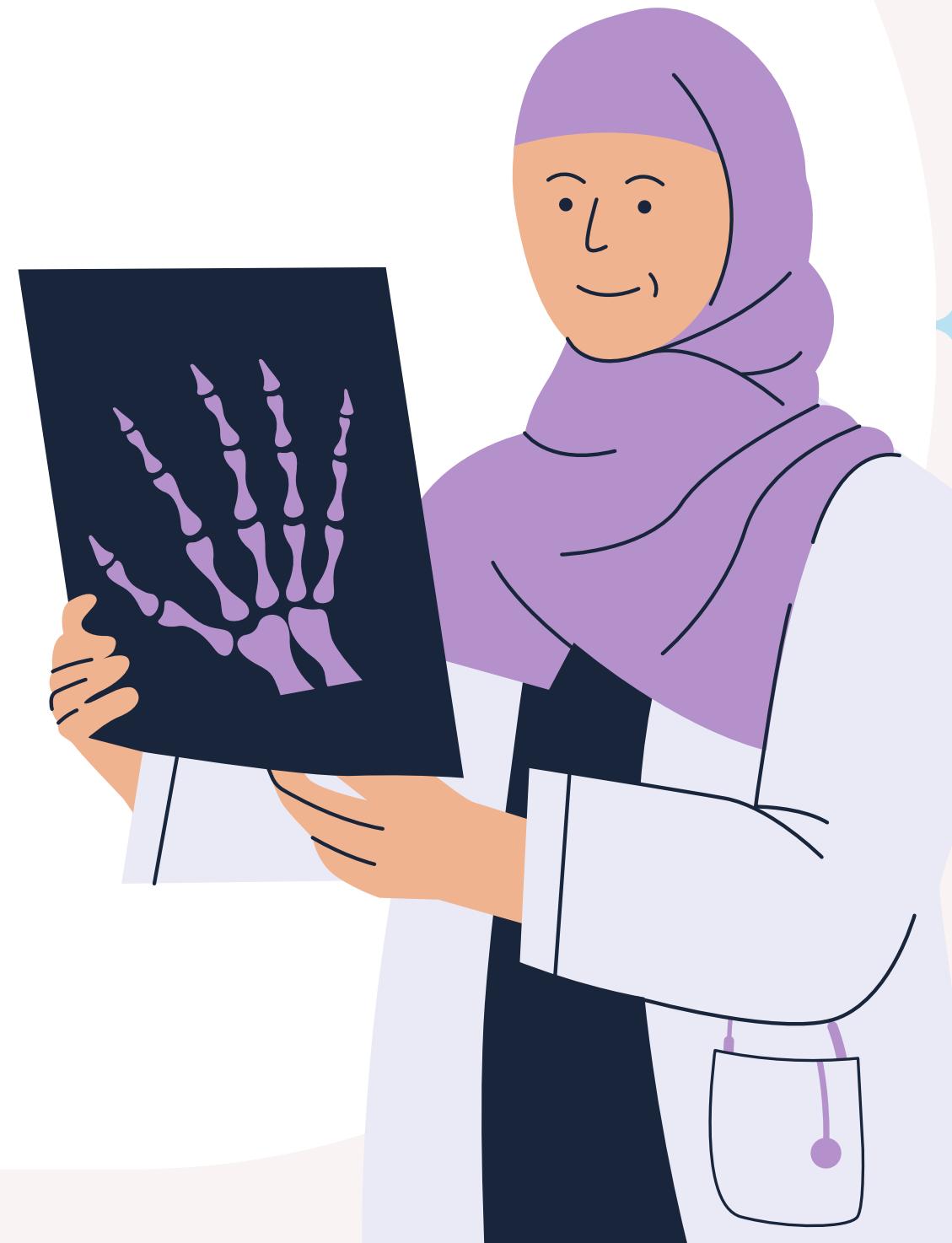
## 1.1 Real-life context

- Hospitals often store patients, doctors, and treatment data in a single unnormalized table (UNF).
- This causes redundancy, inconsistent updates, anomalies when inserting/deleting, and large storage usage.
- Reporting becomes inaccurate and difficult to manage.
- The project aims to normalize data and provide a user-friendly management system.



## 1.2 System Objectives

- Develop a reliable application for managing patients, doctors, treatment categories, and sessions.
- Normalize the database from UNF → 3NF to reduce redundancy and errors.
- Provide a GUI supporting CRUD, search, reporting, and dashboards.
- Enable staff to manage information accurately and make informed decisions.



## 1.3 Scope and Target Users

- Focus on basic hospital data management: input, processing, storage, retrieval.
- Excludes advanced modules: EMR, billing, insurance, authentication, etc.
- Main users: healthcare and administrative staff.
- Suitable for education and demonstration in database and application development courses.

## 1.4 Methods and Technologies Used

- Apply data normalization ( $UNF \rightarrow 3NF$ ), functional dependency analysis, ERD design, and relational schema construction.
- Backend built with Python; GUI using Tkinter (or PyQt/Streamlit alternatives).
- Database: MySQL, connected via mysql-connector/PyMySQL.
- Source code and SQL scripts stored and managed on GitHub for reuse, testing, and deployment.



# Chapter 2

## UNF to 3NF Normalization Process



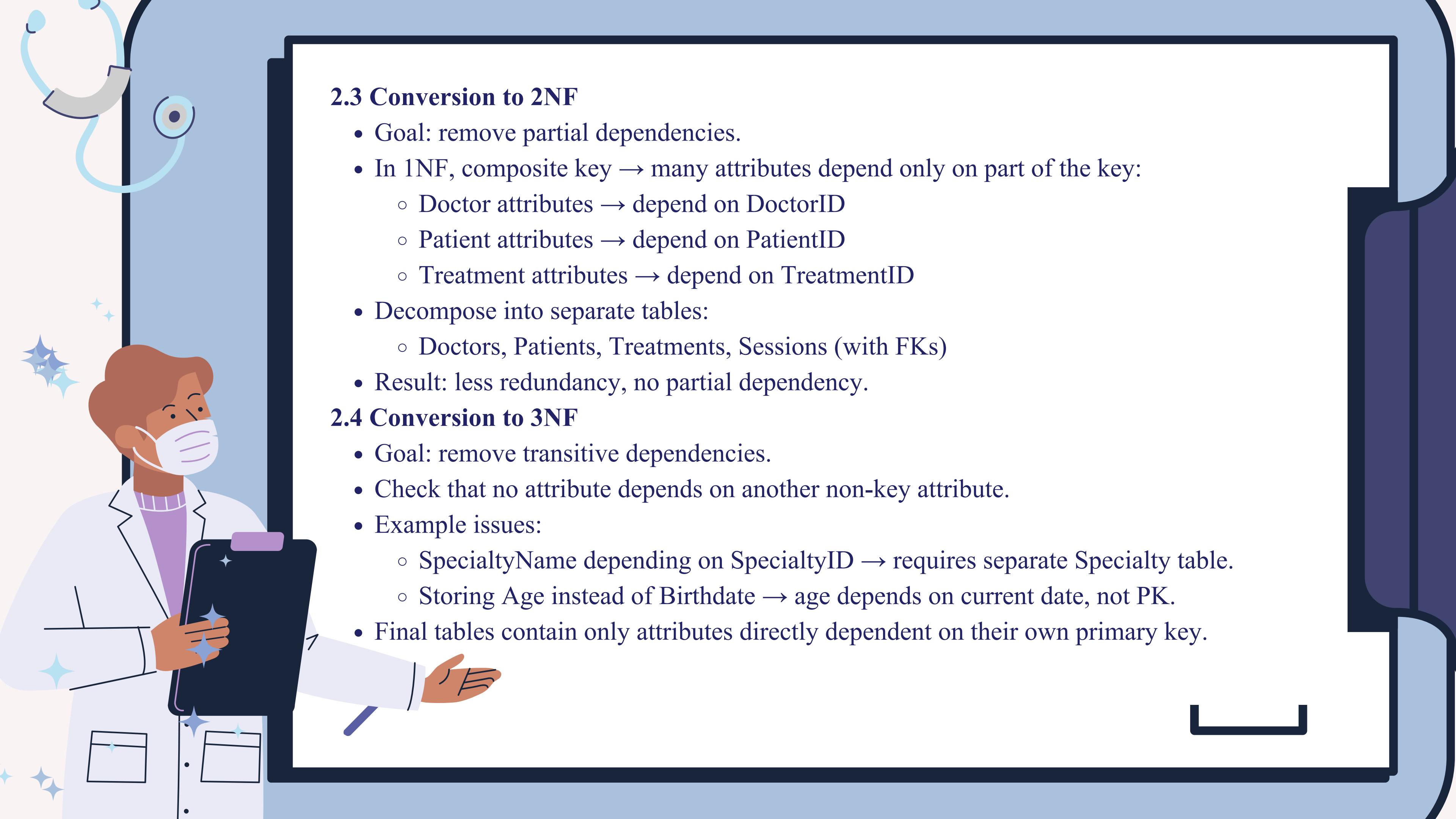
## 2.1 Initial UNF Structure

- All patient, doctor, treatment, and session data stored in one unnormalized table.
- Issues:
  - Redundancy (doctor/patient repeated).
  - Insertion anomaly (cannot add doctor without session).
  - Update anomaly (phone change = update many rows).
  - Deletion anomaly (deleting session may delete doctor/patient info).

## 2.2 Conversion to 1NF

- Ensure atomic values (no grouped or nested data).
- Remove repeating groups → each record = one treatment session.
- Define SessionID (or composite key) to uniquely identify each row.
- Resulting structure contains separate fields for:
  - Patient info
  - Doctor info
  - Treatment info
  - Session info



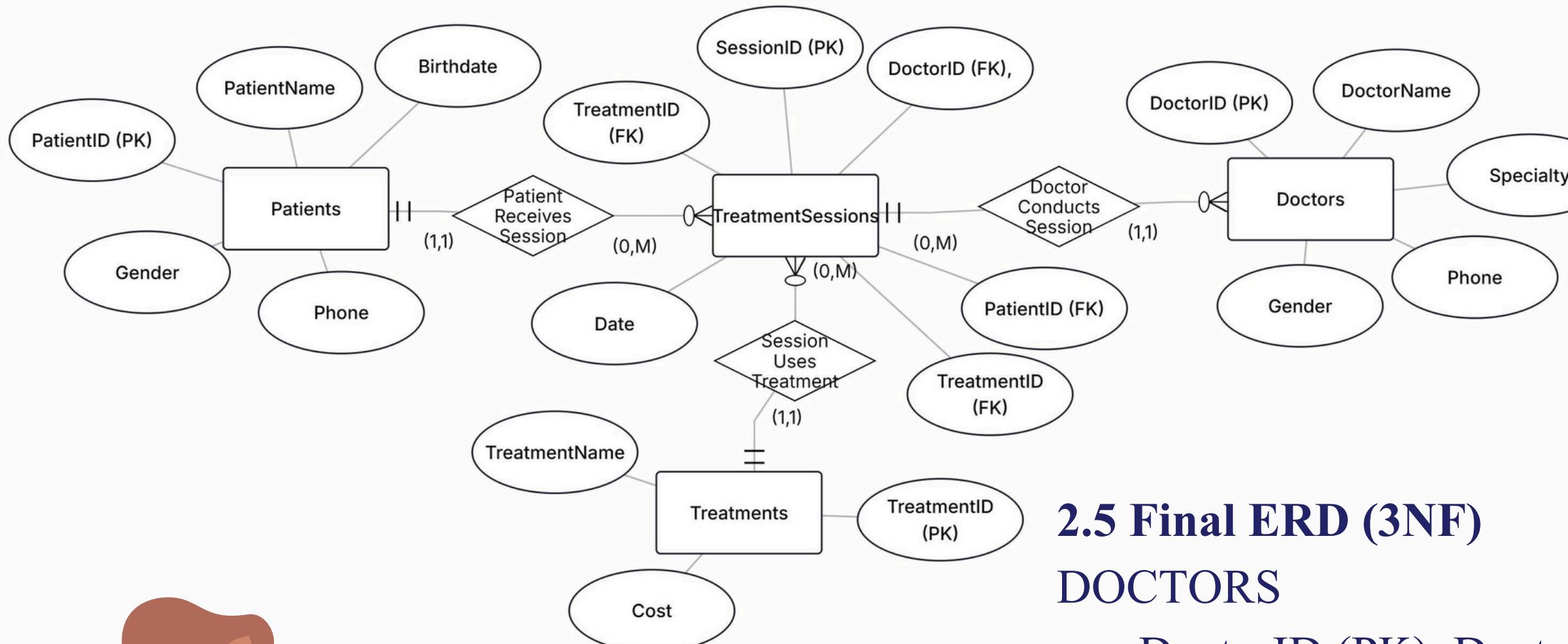


## 2.3 Conversion to 2NF

- Goal: remove partial dependencies.
- In 1NF, composite key → many attributes depend only on part of the key:
  - Doctor attributes → depend on DoctorID
  - Patient attributes → depend on PatientID
  - Treatment attributes → depend on TreatmentID
- Decompose into separate tables:
  - Doctors, Patients, Treatments, Sessions (with FKs)
- Result: less redundancy, no partial dependency.

## 2.4 Conversion to 3NF

- Goal: remove transitive dependencies.
- Check that no attribute depends on another non-key attribute.
- Example issues:
  - SpecialtyName depending on SpecialtyID → requires separate Specialty table.
  - Storing Age instead of Birthdate → age depends on current date, not PK.
- Final tables contain only attributes directly dependent on their own primary key.



## 2.5 Final ERD (3NF)

### DOCTORS

- DoctorID (PK), DoctorName, Gender, Phone, Specialty

### PATIENTS

- PatientID (PK), PatientName, Gender, Birthdate, Phone

### TREATMENTS

- TreatmentID (PK), TreatmentName, Cost

### SESSIONS

- SessionID (PK)

- DoctorID (FK), PatientID (FK), TreatmentID (FK), Date

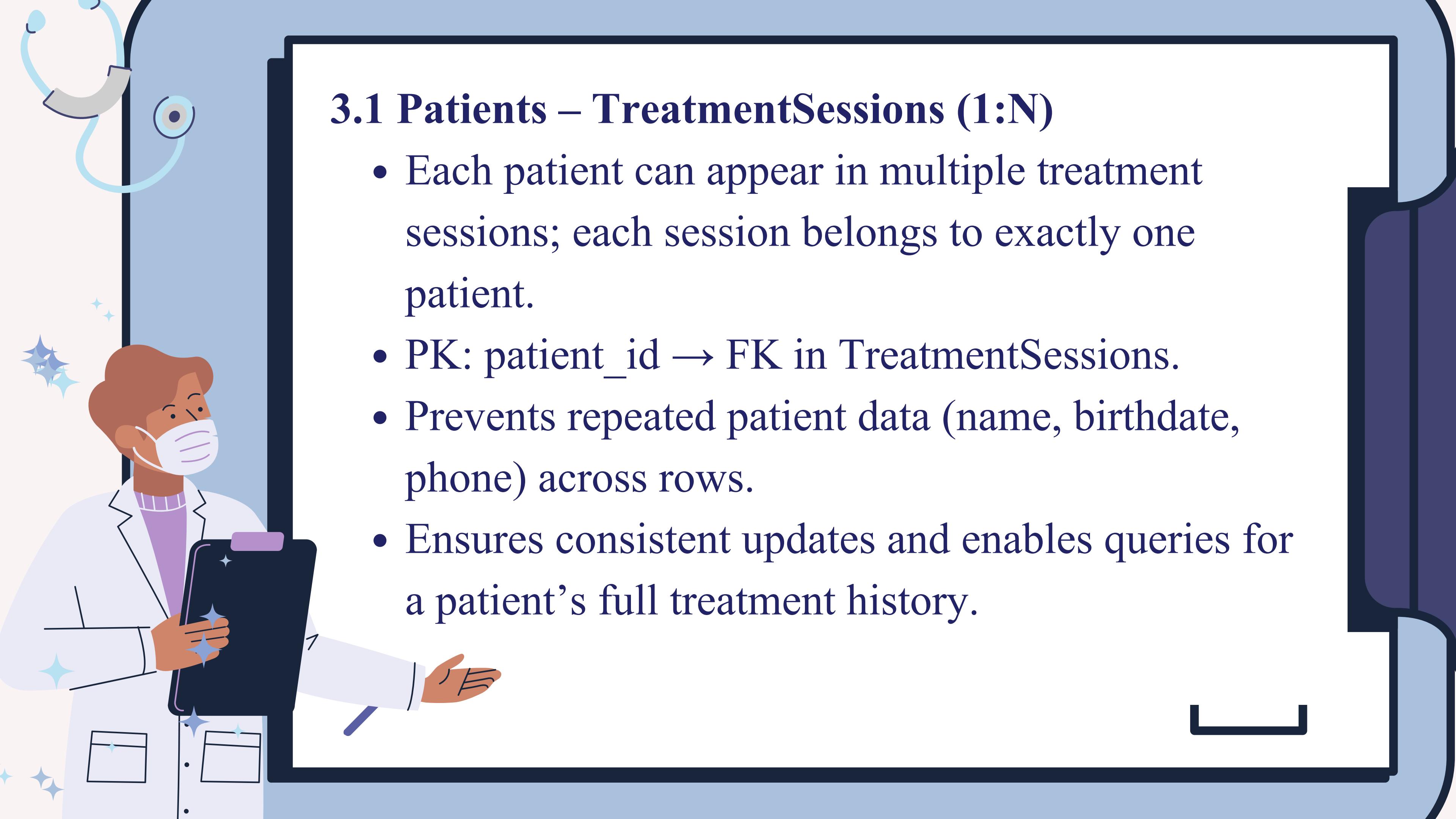
The Sessions table links all entities and enforces referential integrity.

# Chapter 3

## ERD

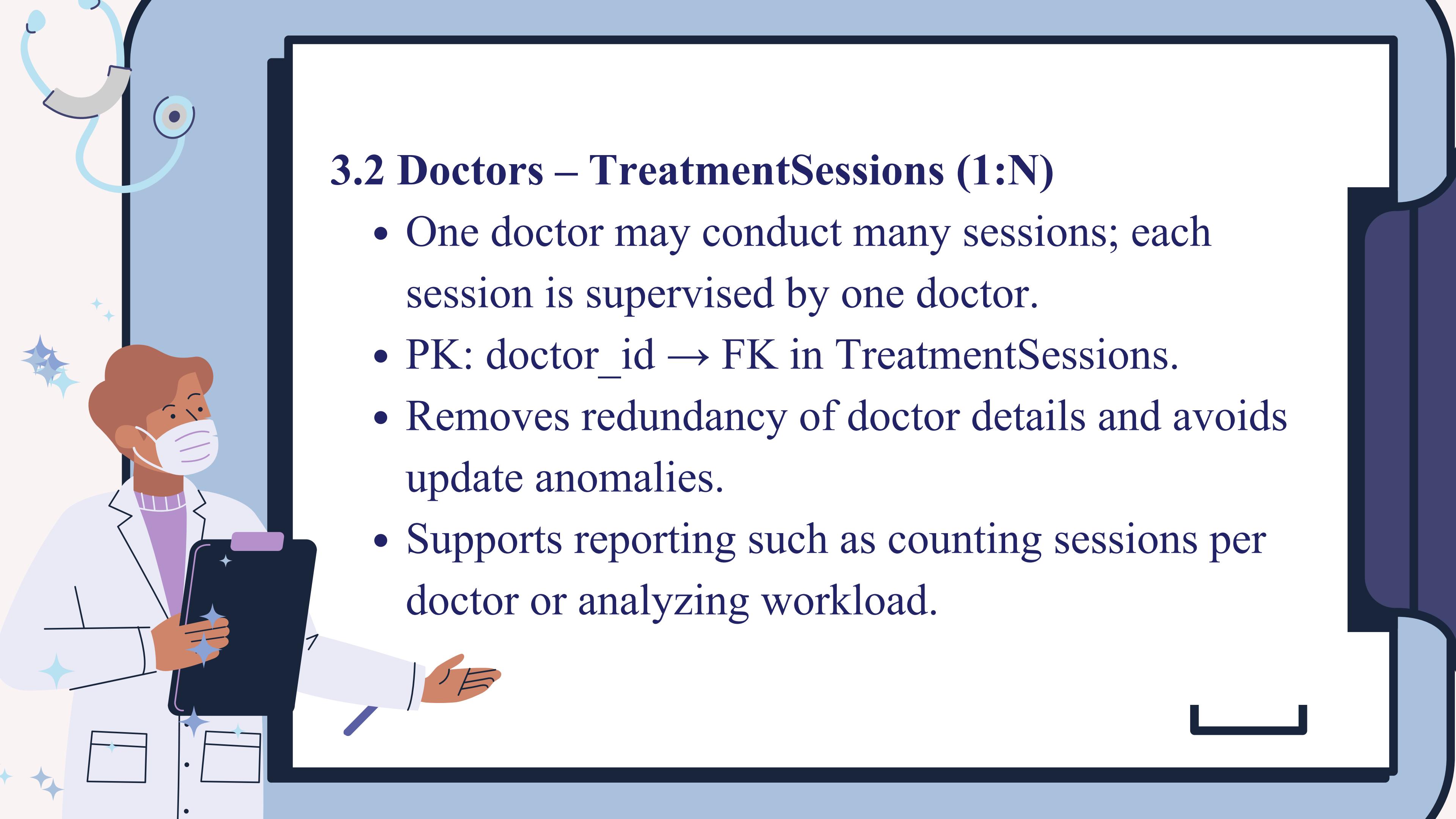
# Relationships





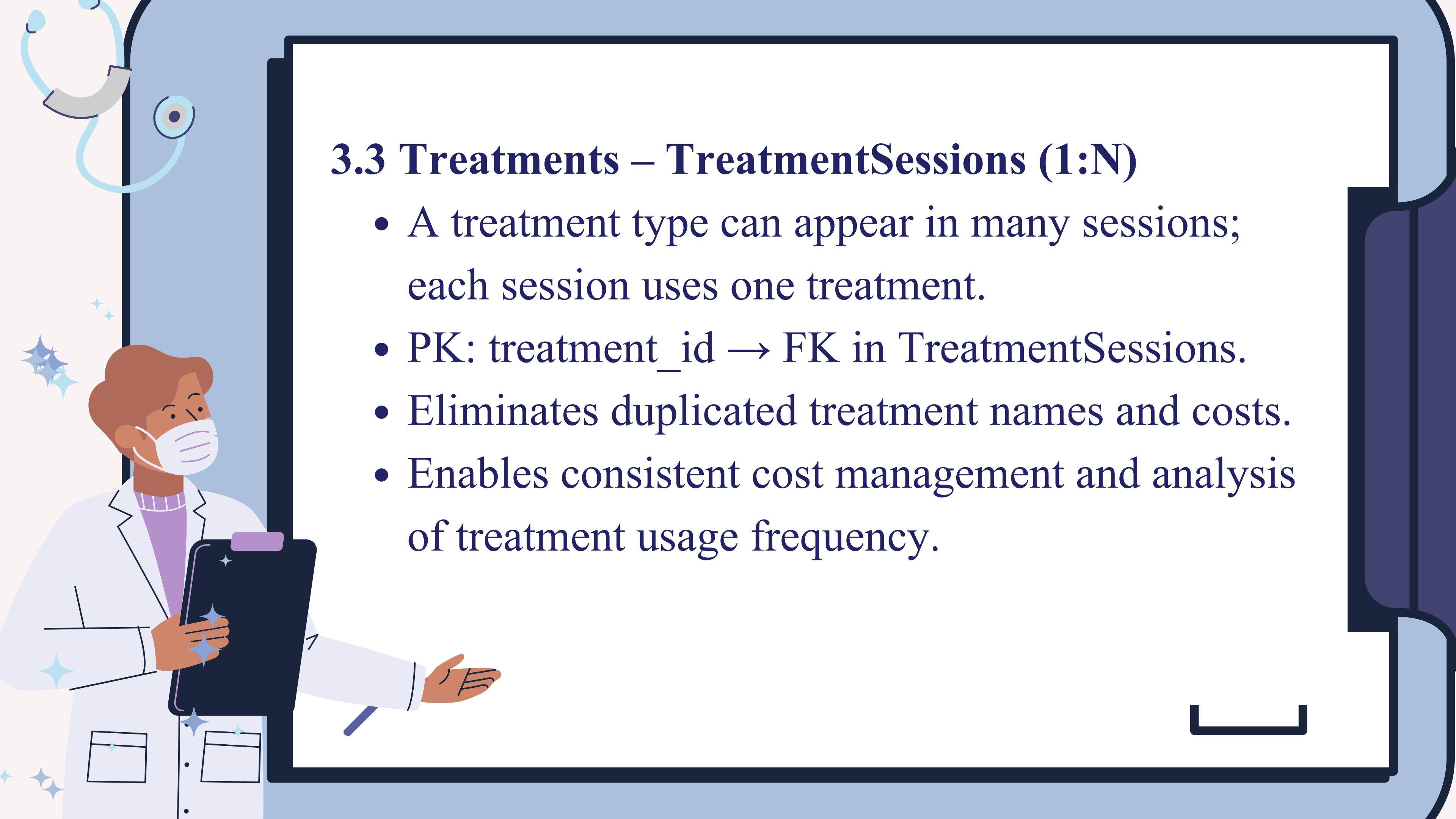
### 3.1 Patients – TreatmentSessions (1:N)

- Each patient can appear in multiple treatment sessions; each session belongs to exactly one patient.
- PK: patient\_id → FK in TreatmentSessions.
- Prevents repeated patient data (name, birthdate, phone) across rows.
- Ensures consistent updates and enables queries for a patient's full treatment history.



## 3.2 Doctors – TreatmentSessions (1:N)

- One doctor may conduct many sessions; each session is supervised by one doctor.
- PK: `doctor_id` → FK in `TreatmentSessions`.
- Removes redundancy of doctor details and avoids update anomalies.
- Supports reporting such as counting sessions per doctor or analyzing workload.



### 3.3 Treatments – TreatmentSessions (1:N)

- A treatment type can appear in many sessions; each session uses one treatment.
- PK: treatment\_id → FK in TreatmentSessions.
- Eliminates duplicated treatment names and costs.
- Enables consistent cost management and analysis of treatment usage frequency.

# Chapter 4

## Database

### Implementation

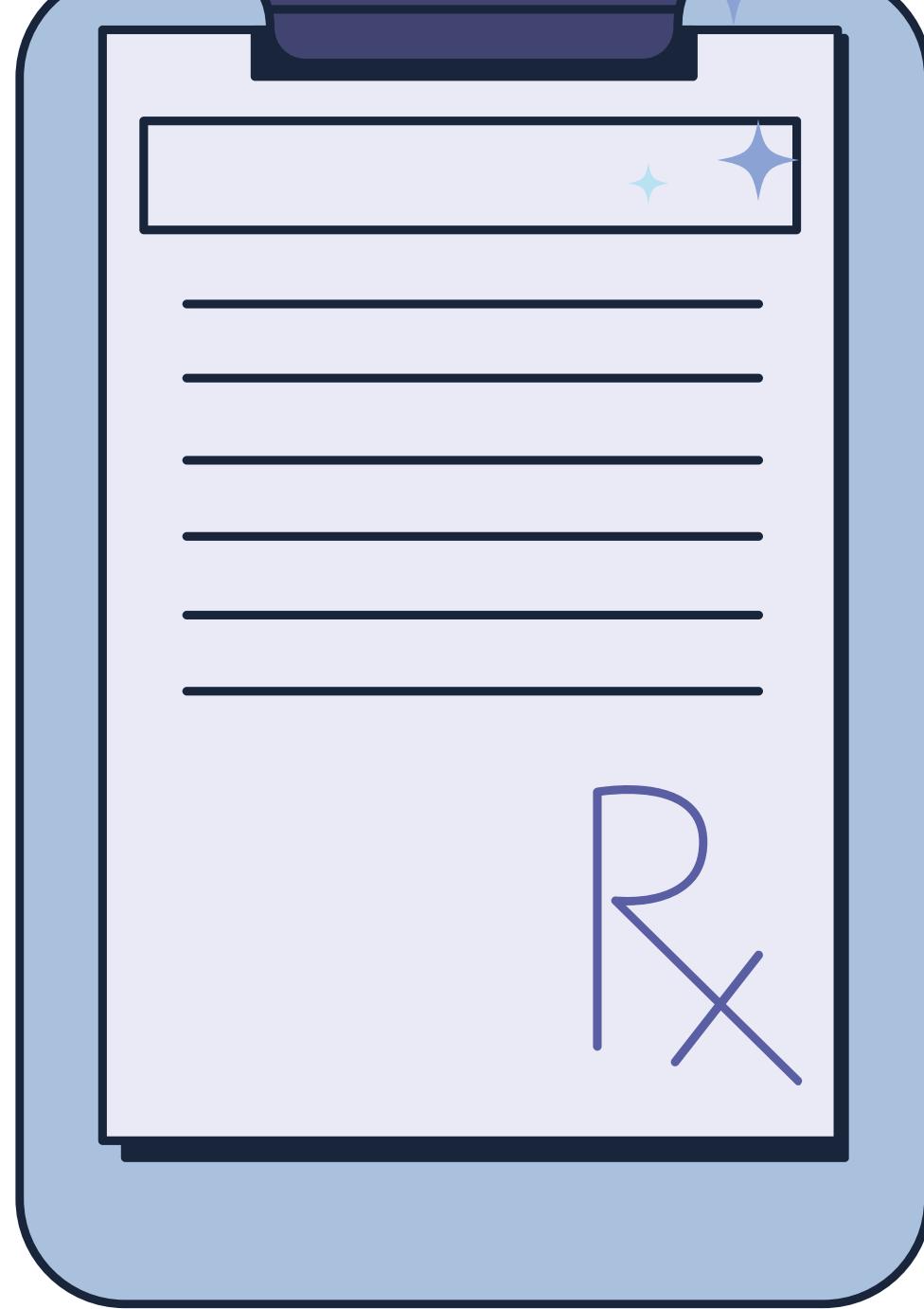


## 4.1 Table Structures

- PATIENTS: PatientID, Name, Birthdate, Gender, Phone.
- DOCTORS: DoctorID, Name, Specialty, Gender, Phone.
- TREATMENTS: TreatmentID, Name, Cost.
- TREATMENTSESSIONS: SessionID + FK to Patient, Doctor, Treatment + Date.

→ Fully normalized (3NF), no redundancy, clear entity separation.





## 4.2 Integrity Constraints

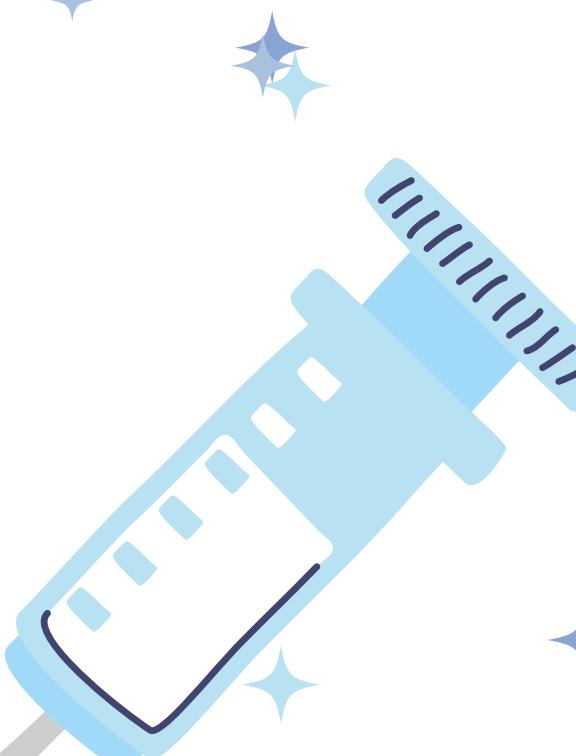
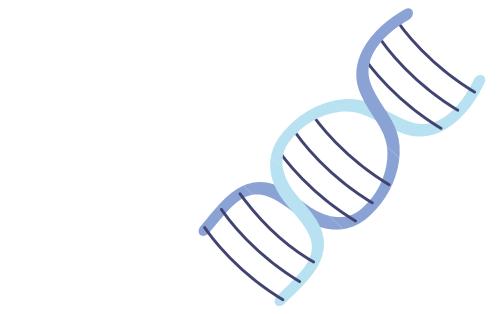
- PKs: PatientID, DoctorID, TreatmentID, SessionID.
- FKS: Enforce valid patient/doctor/treatment references in sessions.
- CHECK: Cost > 0, valid gender, valid phone, valid date.
- UNIQUE: Prevent duplicate names/records where necessary.



## 4.3 Sample Data

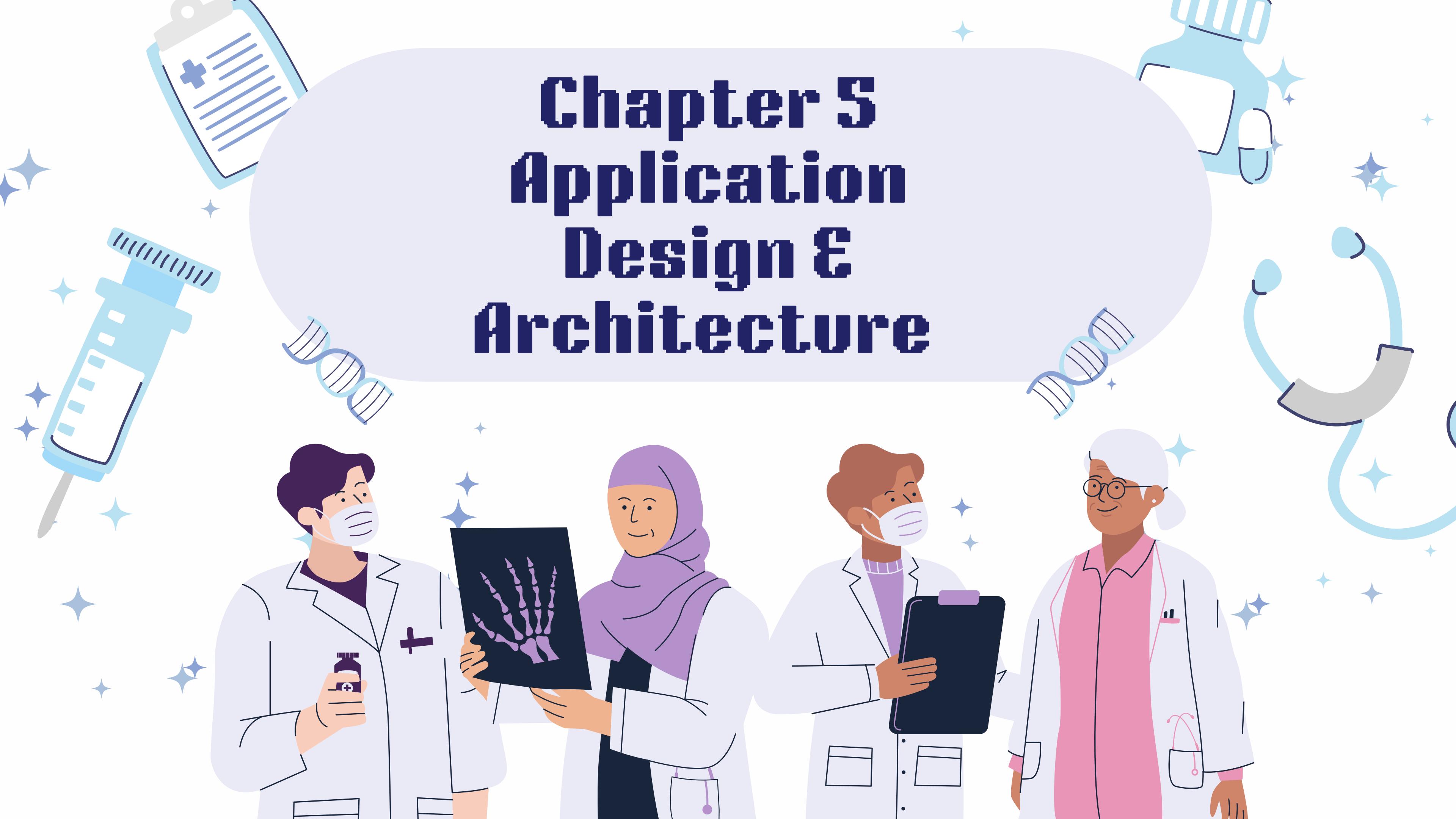
- 50+ patients
- 10+ doctors
- 10+ treatments
- 40–80 sessions

→ Supports CRUD testing, JOIN queries, analytics, and dashboards.



# Chapter 5

## Application Design & Architecture



## 5.1 Technology Stack

- Python 3.14 – core programming language
- MySQL 8.4 – database engine
- Tkinter – GUI framework for desktop
- mysql-connector-python – DB connectivity
- VSCode – development environment
- GitHub – version control & collaboration

→ Provides a lightweight, reliable, and easy-to-maintain system.



## 5.2 Project Structure

```
├── .venv/          # Python virtual environment  
├── .vscode/        # VS Code configuration (optional)  
|   └── settings.json  
├── app/           # Core application source code  
|   ├── db/          # Database interactions and setup  
|   |   ├── connection.py    # Database connection/configuration helper  
|   |   ├── schema.sql     # SQL script for creating tables/structure  
|   |   └── seed.sql       # SQL script for initial data population  
|   ├── dashboard.py    # Logic for the staff dashboard/analytics view  
|   ├── main.py         # Main application/server entry point  
|   ├── queries.py      # Raw or complex database query functions  
|   └── services.py     # Business logic/service layer (e.g., patient check-in  
|       logic)  
└── doc/            # Placeholder for documentation/reports  
    ├── latex.pdf  
    └── slides.pdf      # Placeholder for documentation/reports  
└── requirements.txt # Project dependencies  
└── README.md        # Project overview and setup instructions  
└── .env.example.txt # Example for environment variables (DB credentials,  
                     secret keys)
```



## 5.3 Database Connection Layer

- Centralized MySQL connection module
  - Reusable query executor functions
  - Built-in exception handling
  - Transaction-safe CRUD operations
- Ensures clean separation of logic and stable database access.

## 5.4 User Interface Design

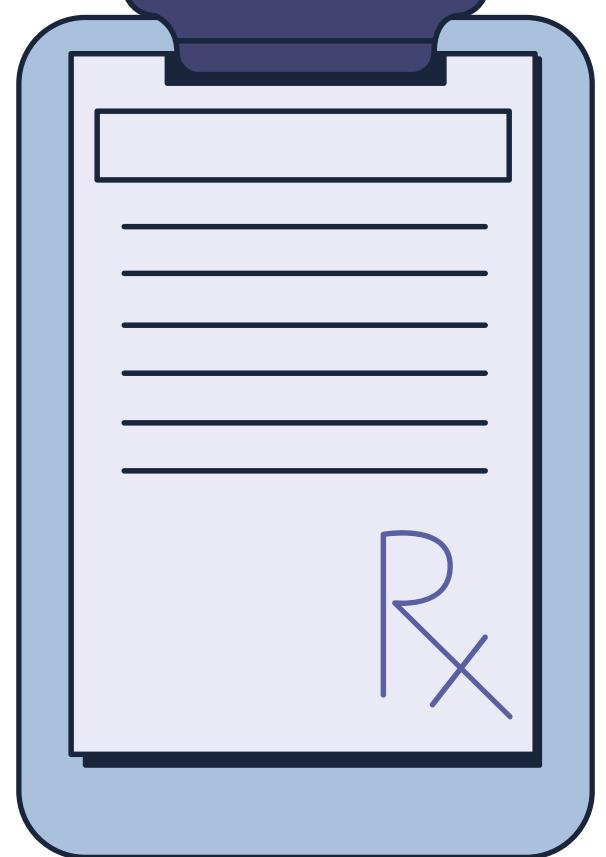
- CRUD Forms (patients, doctors, treatments, sessions) with input validation
  - Data Tables (sortable & scrollable) for record viewing/editing
  - Dashboard & Charts for key statistics
  - Search & Filter tools for fast data lookup
- Intuitive, simple, and optimized for hospital staff.



# Chapter 6

## Functional Features





## 6.1 CRUD Interfaces

The system provides complete CRUD (Create, Read, Update, Delete) functionality for four core data entities. Each interface includes a data-entry form, a data table, validation logic, and database synchronization.



Figure 6.1: CRUD (Create, Read, Update, Delete)



## 6.1.1 Patients

### Features

- Add new patients (Name, Birthdate, Phone, Gender).
- View all patients in a scrollable table.
- Update selected patient info.
- Delete selected patient.
- Clear form / Refresh table.

### Validation

- Birthdate: YYYY-MM-DD.
- Gender via dropdown.
- Phone must be numeric.
- Patient ID checked for duplicates.

Hospital Management System									
Patients	Doctors	Treatments	Sessions	Global Search	Reports	Dashboard			
Patient ID (blank for new): <input type="text" value="1"/>					<input type="button" value="Add"/>	<input type="button" value="Update"/>	<input type="button" value="Delete Selected"/>	<input type="button" value="Clear"/>	<input type="button" value="Refresh"/>
Name:	John Doe				1	John Doe	1985-04-12	0911111111	Male
Birthdate (YYYY-MM-DD):	1985-04-12				2	Jane Smith	1998-11-20	0922222222	Female
Phone:	0911111111				3	Robert Brown	1955-01-01	0933333333	Male
Gender:	Male				4	Nguyen Van A	1989-02-10	0944444444	Male
					5	Tran Thi B	1992-07-20	0955555555	Female
					6	Le Van C	1985-03-15	0966666666	Male
					7	Pham Thi D	1990-11-22	0977777777	Female
					8	Hoang Van E	2001-05-14	0988888888	Male
					9	Ngo Thi F	1999-10-01	0999999999	Female
					10	Phan Van G	1978-01-19	0900000010	Male
					11	Bui Thi H	1982-04-07	0900000011	Female
					12	Vo Van I	1995-12-12	0900000012	Male
					13	Dang Thi J	1998-09-17	0900000013	Female
					14	Nguyen Van K	1983-09-23	0900000014	Male
					15	Tran Thi L	1991-04-04	0900000015	Female
					16	Le Van M	1997-08-09	0900000016	Male
					17	Pham Thi N	1980-11-18	0900000017	Female
					18	Hoang Van O	1979-06-26	0900000018	Male
					19	Ngo Thi P	1987-05-05	0900000019	Female
					20	Phan Van Q	1996-03-29	0900000020	Male
					21	Patient 21	1984-06-10	0900000021	Male
					22	Patient 22	1985-06-11	0900000022	Female
					23	Patient 23	1986-06-12	0900000023	Male

## 6.1.2 Doctors

### Features

- Add new doctor (Name, Phone, Specialty, Gender).
- View all doctors in a scrollable table.
- Update selected doctor info.
- Delete doctor (blocked if referenced in sessions).
- Clear form / Refresh list.

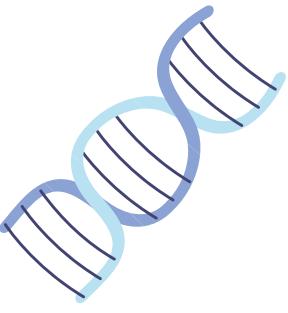
### Validation

- Specialty, Gender selected from dropdowns.
- Phone must be valid and non-empty.
- Doctor ID checked for duplicates.

Hospital Management System

Patients	Doctors	Treatments	Sessions	Global Search	Reports	Dashboard																																																								
Doctor ID (blank for new):	<input type="text" value="1"/>																																																													
Name:	<input type="text"/>																																																													
Specialty:	<input type="text" value="Cardiology"/>																																																													
Phone:	<input type="text" value="0911111111"/>																																																													
Gender:	<input type="text" value="Male"/>																																																													
<input type="button" value="Add"/> <input type="button" value="Update"/> <input type="button" value="Delete Selected"/> <input type="button" value="Clear"/> <input type="button" value="Refresh"/>		<table border="1"> <thead> <tr> <th>Id</th> <th>Name</th> <th>Specialty</th> <th>Phone</th> <th>Gender</th> </tr> </thead> <tbody> <tr><td>1</td><td>Dr. Alistair Finch</td><td>Cardiology</td><td>0911111111</td><td>Male</td></tr> <tr><td>2</td><td>Dr. Evelyn Reed</td><td>Pediatrics</td><td>0922222222</td><td>Female</td></tr> <tr><td>3</td><td>Dr. Marcus Cole</td><td>Orthopedics</td><td>0933333333</td><td>Male</td></tr> <tr><td>4</td><td>Dr. Sofia Perez</td><td>General Practice</td><td>0944444444</td><td>Female</td></tr> <tr><td>5</td><td>Dr. Minh</td><td>Cardiology</td><td>0955555555</td><td>Male</td></tr> <tr><td>6</td><td>Dr. Hoa</td><td>Neurology</td><td>0966666666</td><td>Female</td></tr> <tr><td>7</td><td>Dr. Truong</td><td>Oncology</td><td>0977777777</td><td>Male</td></tr> <tr><td>8</td><td>Dr. Hanh</td><td>Dermatology</td><td>0988888888</td><td>Female</td></tr> <tr><td>9</td><td>Dr. Duc</td><td>Orthopedic</td><td>0999999999</td><td>Male</td></tr> <tr><td>10</td><td>Dr. Quynh</td><td>Pediatrics</td><td>0900000010</td><td>Female</td></tr> </tbody> </table>						Id	Name	Specialty	Phone	Gender	1	Dr. Alistair Finch	Cardiology	0911111111	Male	2	Dr. Evelyn Reed	Pediatrics	0922222222	Female	3	Dr. Marcus Cole	Orthopedics	0933333333	Male	4	Dr. Sofia Perez	General Practice	0944444444	Female	5	Dr. Minh	Cardiology	0955555555	Male	6	Dr. Hoa	Neurology	0966666666	Female	7	Dr. Truong	Oncology	0977777777	Male	8	Dr. Hanh	Dermatology	0988888888	Female	9	Dr. Duc	Orthopedic	0999999999	Male	10	Dr. Quynh	Pediatrics	0900000010	Female
Id	Name	Specialty	Phone	Gender																																																										
1	Dr. Alistair Finch	Cardiology	0911111111	Male																																																										
2	Dr. Evelyn Reed	Pediatrics	0922222222	Female																																																										
3	Dr. Marcus Cole	Orthopedics	0933333333	Male																																																										
4	Dr. Sofia Perez	General Practice	0944444444	Female																																																										
5	Dr. Minh	Cardiology	0955555555	Male																																																										
6	Dr. Hoa	Neurology	0966666666	Female																																																										
7	Dr. Truong	Oncology	0977777777	Male																																																										
8	Dr. Hanh	Dermatology	0988888888	Female																																																										
9	Dr. Duc	Orthopedic	0999999999	Male																																																										
10	Dr. Quynh	Pediatrics	0900000010	Female																																																										





Hospital Management System

## Hospital Management System

- Patients
- Doctors
- Treatments**
- Sessions
- Global Search
- Reports
- Dashboard

Treatment ID (blank for new):

Name:

Cost (e.g. 150.00):

**Add** **Update** **Delete Selected** **Clear** **Refresh**

Id	Name	Cost
1	Initial Consultation	150.0
2	X-Ray Scan	75.5
3	Physical Therapy	90.0
4	Echocardiogram	320.0
5	MRI Scan	1200.0
6	Chemotherapy	2500.0
7	Skin Treatment	800.0
8	Blood Test	150.0
9	Ultrasound	450.0
10	CT Scan	1500.0
12	Vaccination	200.0

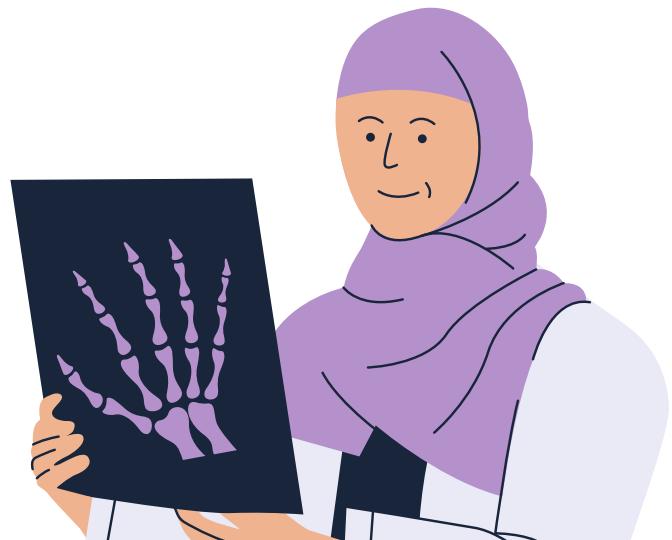
### 6.1.3 Treatments

#### Features

- Add new treatment (Name, Cost).
- View all treatments (Id, Name, Cost).
- Update selected treatment.
- Delete treatment (blocked if used in sessions).
- Clear form / Refresh treatment list.

#### Validation

- Name must not be empty.
- Cost must be a positive number.
- Treatment ID checked for duplicates.



management System

## Hospital Management System

- Doctors
- Treatments
- Sessions**
- Global Search
- Reports
- Dashboard

blank for new):

1	1: John Doe	4: Dr. Sofia Perez	1: Initial Consultation	2024-10-01
2	1: John Doe	1: Dr. Alistair Finch	4: Echocardiogram	2024-10-15
3	2: Jane Smith	2: Dr. Evelyn Reed	1: Initial Consultation	2024-11-05
4	3: Robert Brown	3: Dr. Marcus Cole	2: X-Ray Scan	2024-11-10
5	4: Nguyen Van A	1: Dr. Alistair Finch	5: MRI Scan	2024-01-05
6	5: Tran Thi B	2: Dr. Evelyn Reed	2: X-Ray Scan	2024-01-06
7	6: Le Van C	3: Dr. Marcus Cole	3: Physical Therapy	2024-01-07
8	7: Pham Thi D	4: Dr. Sofia Perez	1: Initial Consultation	2024-01-08
9	8: Hoang Van E	5: Dr. Minh	7: Skin Treatment	2050-12-30
10	9: Ngo Thi F	6: Dr. Hoa	12: Vaccination	2024-01-10
11	10: Phan Van G	7: Dr. Truong	8: Blood Test	2024-01-11
12	11: Bui Thi H	8: Dr. Hanh	6: Chemotherapy	2024-01-12
13	12: Vo Van I	9: Dr. Duc	10: CT Scan	2024-01-13
14	13: Dang Thi J	10: Dr. Quynh	9: Ultrasound	2024-01-14
15	14: Nguyen Van K	1: Dr. Alistair Finch	4: Echocardiogram	2024-01-15
16	15: Tran Thi L	2: Dr. Evelyn Reed	5: MRI Scan	2024-01-16
17	16: Le Van M	3: Dr. Marcus Cole	11: Physiotherapy	2024-01-17
18	17: Pham Thi N	4: Dr. Sofia Perez	2: X-Ray Scan	2024-01-18
19	18: Hoang Van O	5: Dr. Minh	3: Physical Therapy	2024-01-19
20	19: Ngo Thi P	6: Dr. Hoa	1: Initial Consultation	2024-01-20
21	20: Phan Van Q	7: Dr. Truong	8: Blood Test	2024-01-21
22	21: Patient 21	8: Dr. Hanh	4: Echocardiogram	2024-01-22
23	22: Patient 22	9: Dr. Duc	6: Chemotherapy	2024-01-23

-MM-DD):

2024-10-01

**Buttons:** Update, Delete Selected, Clear, Refresh

## 6.1.4 Treatment Sessions

### Features

- Add new session (select Patient, Doctor, Treatment + Date).
- View all sessions (Id, Patient, Doctor, Treatment, Date).
- Update selected session.
- Delete session (FK rules enforced).
- Clear form / Refresh session list.

### Validation

- All dropdown selections are required.
- Date must follow YYYY-MM-DD.
- Prevents incomplete or invalid session entries.

## 6.2 Search & Filter Mechanisms

### Global Search

- A dedicated tab inside the Sessions module.
- Users enter a keyword to search across:
  - Patient names
  - Doctor names
  - Treatment names

### Results

- Displayed in a unified table with session ID, patient, doctor, treatment, cost, and date.
- Supports both vertical and horizontal scrolling.

### Benefits

- Fast information retrieval
- No need to switch between multiple CRUD screens
- Useful for reception, verification, and quick history lookup
- Provides a full view of all related session data



Hospital Management System

Hospital Management System

Patients Doctors Treatments Sessions Global Search Reports Dashboard

Global Search (Patient/Doctor/Treatment Name): Minh

session_id	Patient_Name	Doctor_Name	Treatment_Name	Cost	Date
9	Hoang Van E	Dr. Minh	Skin Treatment	800.00	2050-12-30
29	Patient 28	Dr. Minh	X-Ray Scan	75.50	2024-01-29
19	Hoang Van O	Dr. Minh	Physical Therapy	90.00	2024-01-19

Explain more about some facts related to the melting of arctic ice here

The screenshot shows a Windows application window titled "Hospital Management System". The menu bar includes "File", "Edit", "View", "Tools", "Help", and "About". The main menu has tabs for "Patients", "Doctors", "Treatments", "Sessions", "Global Search" (which is selected and highlighted in blue), "Reports", and "Dashboard". Below the menu is a search bar with the placeholder "Global Search (Patient/Doctor/Treatment Name):" followed by a text input field containing "Minh" and a "Search Sessions" button. The main content area displays a table with six columns: session\_id, Patient\_Name, Doctor\_Name, Treatment\_Name, Cost, and Date. The table contains three rows of data. A vertical scroll bar is visible on the right side of the table. At the bottom of the window is a status bar.

session_id	Patient_Name	Doctor_Name	Treatment_Name	Cost	Date
9	Hoang Van E	Dr. Minh	Skin Treatment	800.00	2050-12-30
29	Patient 28	Dr. Minh	X-Ray Scan	75.50	2024-01-29
19	Hoang Van O	Dr. Minh	Physical Therapy	90.00	2024-01-19

Figure 6.6: global search

## 6.3 Required Analytical Queries

### Inner Join Report – Sessions by Patient/Treatment

Shows only patients who have completed treatment sessions, including treatment type, date, and cost. Useful for reviewing completed treatment histories.

	A	B	C	D
1	Patient	Treatment	Date	Cost
2	Hoang Van	Skin Treatn	12/30/2050	800
3	Robert Bro	X-Ray Sca	11/10/2024	75.5
4	John Doe	Echocardio	10/15/2024	320
5	Patient 28	X-Ray Sca	1/29/2024	75.5
6	Patient 27	Vaccination	1/28/2024	200
7	Patient 26	CT Scan	1/27/2024	1500
8	Patient 25	Ultrasound	1/26/2024	450
9	Patient 24	MRI Scan	1/25/2024	1200
10	Patient 23	Skin Treatn	1/24/2024	800
11	Patient 22	Chemothera	1/23/2024	2500

Figure 6.7: Inner Join query result.

## 6.3 Required Analytical Queries

### Left Join Report – All Patients (With/Without Sessions)

Lists all patients, including those with no treatment sessions. Empty fields indicate no recorded session. Helps identify untreated or pending-appointment patients.

	A	B	C	D
1	Patient	Treatment	Date	Cost
2	John Doe	None	10/1/2024	None
3	John Doe	Echocardio	10/15/2024	320
4	Jane Smith	None	11/5/2024	None
5	Robert Bro	X-Ray Sca	11/10/2024	75.5
6	Nguyen Va	MRI Scan	1/5/2024	1200
7	Tran Thi B	X-Ray Sca	1/6/2024	75.5
8	Le Van C	Physical Th	1/7/2024	90
9	Pham Thi D	None	1/8/2024	None
10	Hoang Van	Skin Treatn	12/30/2050	800
11	Ngo Thi F	Vaccination	1/10/2024	200

Figure 6.8: Left Join query result.

## 6.3 Required Analytical Queries

### Multi-Table Join Report – Sessions (Patients, Doctors, Treatments)

Combines patient, doctor, and treatment data to show full session details: who was treated, by which doctor, and when.

	A	B	C	D	E
1	Patient	Doctor	Treatment	Date	Cost
2	Hoang Van	Dr. Minh	Skin Treatn	12/30/2050	800
3	Robert Bro	Dr. Marcus	X-Ray Sca	11/10/2024	75.5
4	John Doe	Dr. Alistair	Echocardio	10/15/2024	320
5	Patient 28	Dr. Minh	X-Ray Sca	1/29/2024	75.5
6	Patient 27	Dr. Sofia P	Vaccination	1/28/2024	200
7	Patient 26	Dr. Marcus	CT Scan	1/27/2024	1500
8	Patient 25	Dr. Evelyn	Ultrasound	1/26/2024	450
9	Patient 24	Dr. Alistair	MRI Scan	1/25/2024	1200
10	Patient 23	Dr. Quynh	Skin Treatn	1/24/2024	800
11	Patient 22	Dr. Duc	Chemother	1/23/2024	2500

Figure 6.9: Multi-table Join result

## 6.3 Required Analytical Queries

### Subquery/Aggregation Report – High-Cost Treatment Patients

Shows treatments costing above the overall average and the patients who received them. Useful for financial monitoring and budgeting.

	A	B	C	D
1	Treatment	Cost	Patient	
2	Chemotherapy	2500	Bui Thi H	
3	Chemotherapy	2500	Patient 22	
4	CT Scan	1500	Vo Van I	
5	CT Scan	1500	Patient 26	
6	MRI Scan	1200	Nguyen Van A	
7	MRI Scan	1200	Tran Thi L	
8	MRI Scan	1200	Patient 24	
9	Skin Treatn	800	Hoang Van E	
10	Skin Treatn	800	Patient 23	

Figure 6.10: High-cost treatments query result.



## Export Report CSV

All reports can be exported to CSV files, enabling offline storage, sharing, and extended analysis outside the application.

Hospital Management System

Hospital Management System

Patient	Treatment	Date	Cost
Hoang Van E	Skin Treatment	2050-12-30	800.0
Robert Brown	X-Ray Scan	2024-11-10	75.5
Jane Smith	Initial Consultation	2024-11-05	150.0
John Doe	Echocardiogram	2024-10-15	320.0
John Doe	Initial Consultation	2024-10-01	150.0
Patient 29	Physiotherapy	2024-01-30	-100.0
Patient 28	X-Ray Scan	2024-01-29	75.5
Patient 27	Vaccination	2024-01-28	200.0
Patient 26	CT Scan	2024-01-27	1500.0
Patient 25	Ultrasound	2024-01-26	450.0
Patient 24	MRI Scan	2024-01-25	1200.0
Patient 23	Skin Treatment	2024-01-24	800.0
Patient 22	Chemotherapy	2024-01-23	2500.0
Patient 21	Echocardiogram	2024-01-22	320.0
Phan Van Q	Blood Test	2024-01-21	150.0
Ngo Thi P	Initial Consultation	2024-01-20	150.0
Hoang Van O	Physical Therapy	2024-01-19	90.0
Pham Thi N	X-Ray Scan	2024-01-18	75.5
Le Van M	Physiotherapy	2024-01-17	-100.0
Tran Thi L	MRI Scan	2024-01-16	1200.0

Figure 6.11: Report

## 6.4 Dashboard Visualizations

The dashboard displays key hospital metrics and a treatment-cost distribution chart.

- KPIs (left): total patients, total doctors, total sessions, average treatment cost, and count of high-cost treatments.
- Histogram (right): shows how treatment costs are distributed, helping identify common ranges and outliers.

This view provides an instant overview of hospital activity and cost patterns.

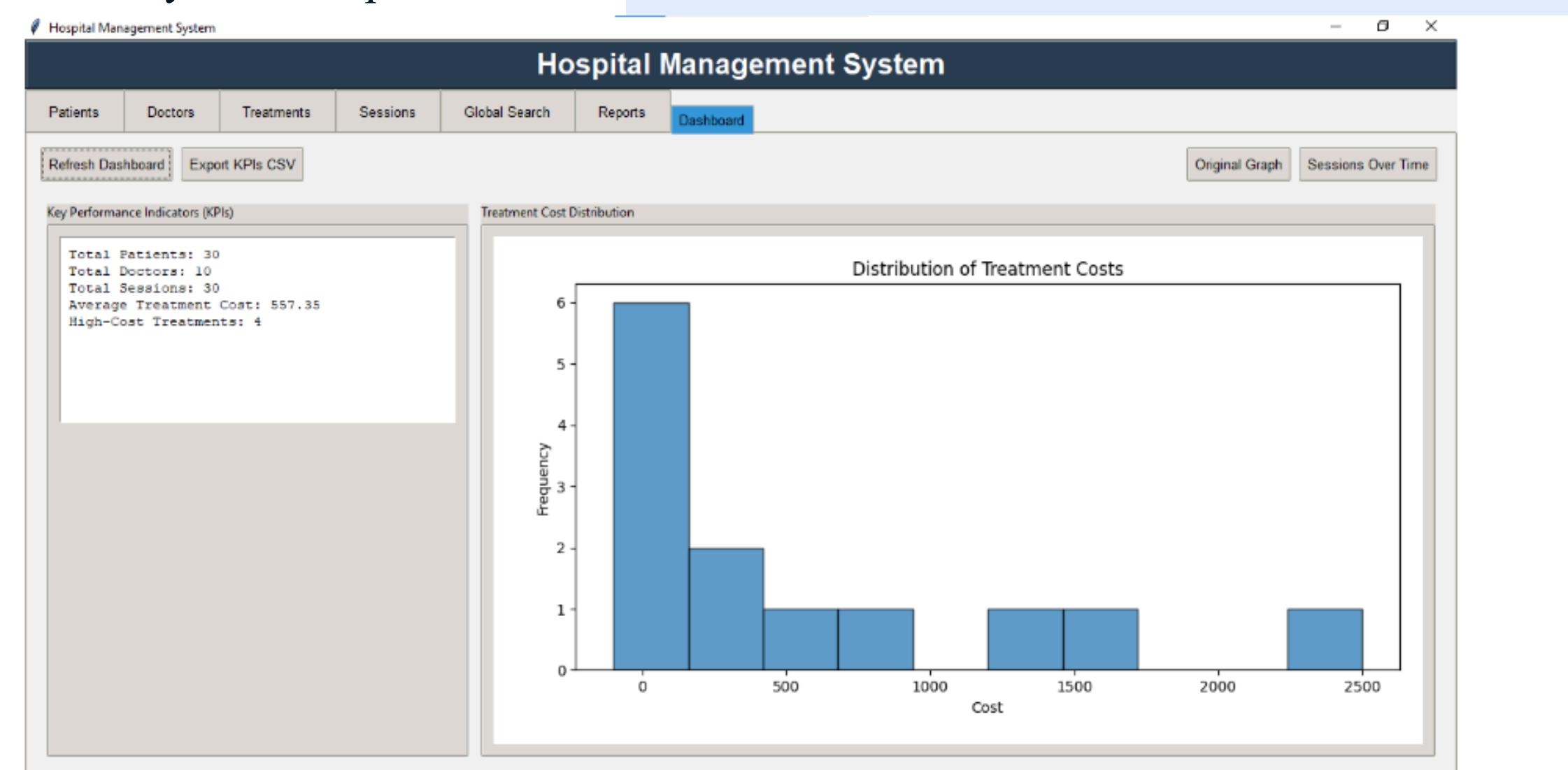
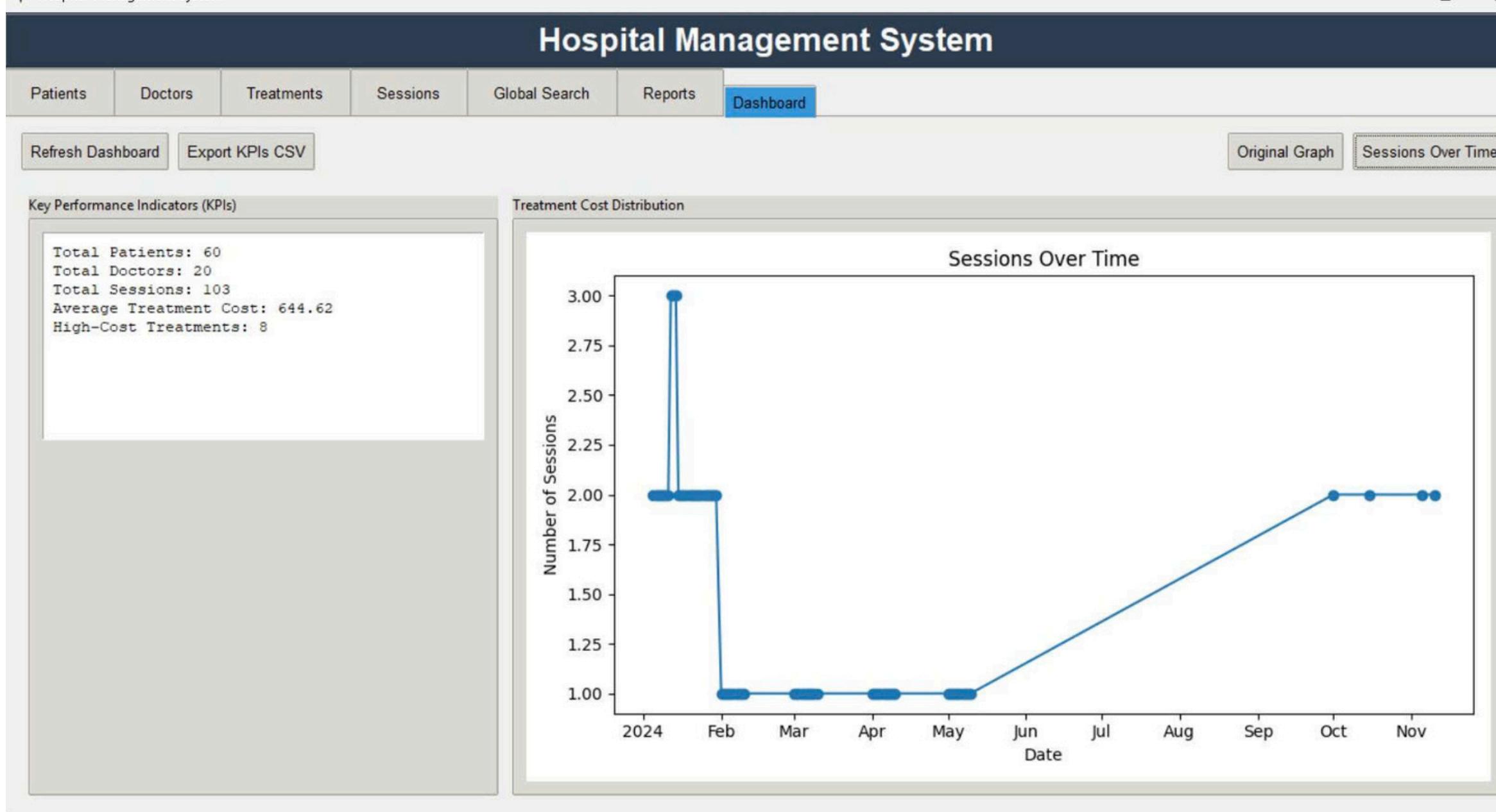


Figure 6.12: KPIs and Treatment Cost Distribution



The figure shows the Dashboard in Sessions Over Time mode, where a line chart displays session counts by date to reveal activity trends. KPIs remain visible on the left for quick reference.

## Dashboard Buttons

- Refresh: Update KPIs and charts
- Export CSV: Download KPI data
- Original Graph: Return to cost distribution
- Sessions Over Time: Switch to time-based chart

# Chapter 7

## Testing and Reliability





## 7.1 Input Validation

- Required fields must be filled.
- Date must be valid; cost must be a positive number.
- Doctor/Treatment selected from dropdown.
- Prevent duplicate IDs and duplicate sessions.
- Foreign keys enforced to keep data consistent.

## 7.2 Error Handling

- Invalid input → warning, no saving.
- DB errors shown with readable messages; app stays responsive.
- Unexpected errors logged.
- Fail-safe mode if DB is unreachable.

## 7.3 Sample Tests

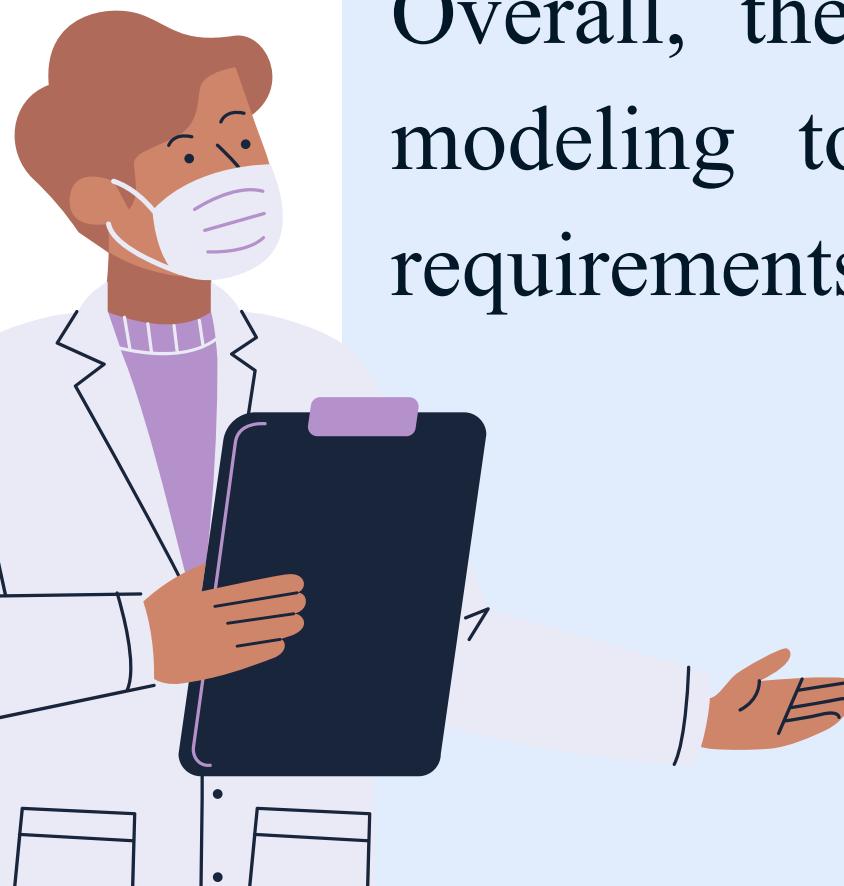
- Wrong date → warning, not saved.
- Letters in cost → blocked.
- Empty search keyword → no results.

# Chapter 8

## Results

## Discussion





The Hospital Patient Manager system integrates a normalized 3NF database, SQL queries, and a Python GUI into a unified workflow.

CRUD operations for Patients, Doctors, Treatments, and Sessions run reliably with strong validation and foreign-key rules.

Analytical queries and dashboard charts provide useful insights on treatment activity, doctor workload, and cost patterns.

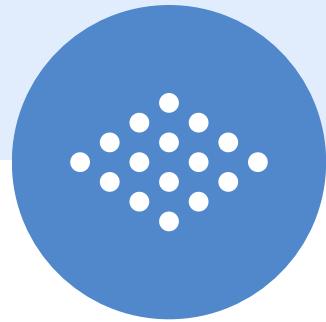
Overall, the system forms a complete pipeline—from data modeling to interface and reporting—fulfilling all project requirements.



# Conclusion

The Hospital Patient Manager system integrates a 3NF database, MySQL backend, and Python/Tkinter GUI into a reliable application. It supports full CRUD operations, strong validation, analytical queries, and dashboard insights. The project meets all requirements and provides a solid foundation for future upgrades such as authentication, web deployment, and advanced analytics.





**THANKS  
FOR  
WATCHING!**