# Department Of Computer Science and Engineering

## Assignment-1:

## Implement a Basic Knowledgebase of Your Choice Using Prolog

**Course Title:** Artificial Intelligence and Expert Systems Lab

**Lab Course Code:** CSE 404

**Submitted To:**

Noor Mairukh Khan Arnob,

Lecturer

Department Of Computer Science & Engineering

**Submitted By:**

Sabber Hossen Shuvo

Reg. : 21201087

Roll: 87

Sec: B2

## i. Problem Title: Hospital & Patients Database Management System

**ii. Problem Description:** This Prolog program represents a basic Hospital Management System for storing and retrieving information about doctors, patients, appointments, surgeries, room assignments, bills, emergency cases, ambulance services, hospital staff, and financial calculations. The system organizes hospital-related data and provides functionalities such as tracking doctor specialties, scheduling appointments, managing room assignments, tracking surgeries, calculating patient bills, handling emergency cases, and providing ambulance services. It also maintains staff records and financial information such as total earnings, salaries, and profits.

**Key Features:**

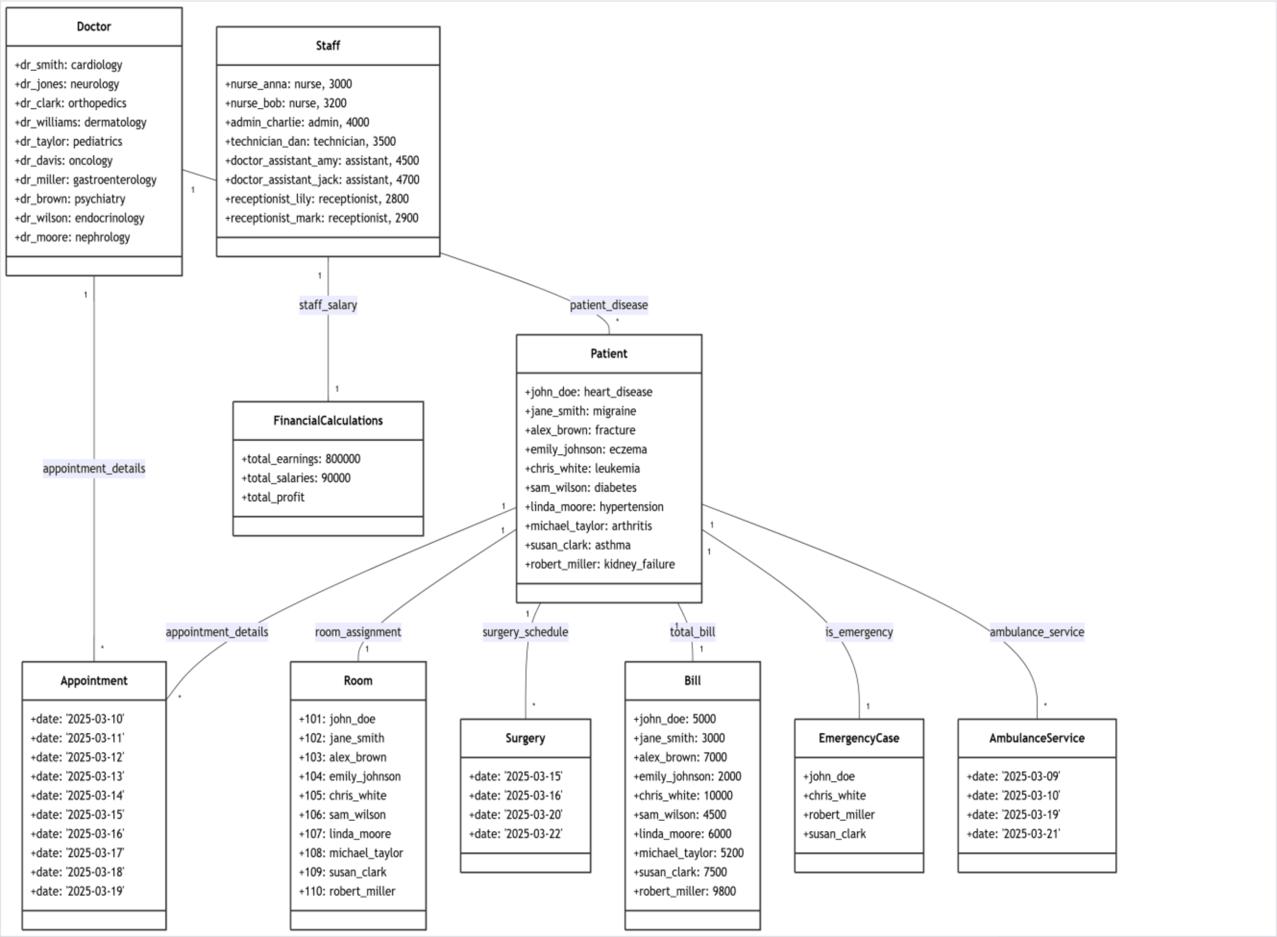
- 1. Doctors & Specialties
- 2. Patients & Illnesses
- 3. Appointments System
- 4. Room Assignments
- 5. Surgery Schedules
- 6. Billing System
- 7. Emergency Cases
- 8. Ambulance Services
- 9. Hospital Staff & Salaries
- 10. Financial Calculations

## iii. Tools and Languages Used:

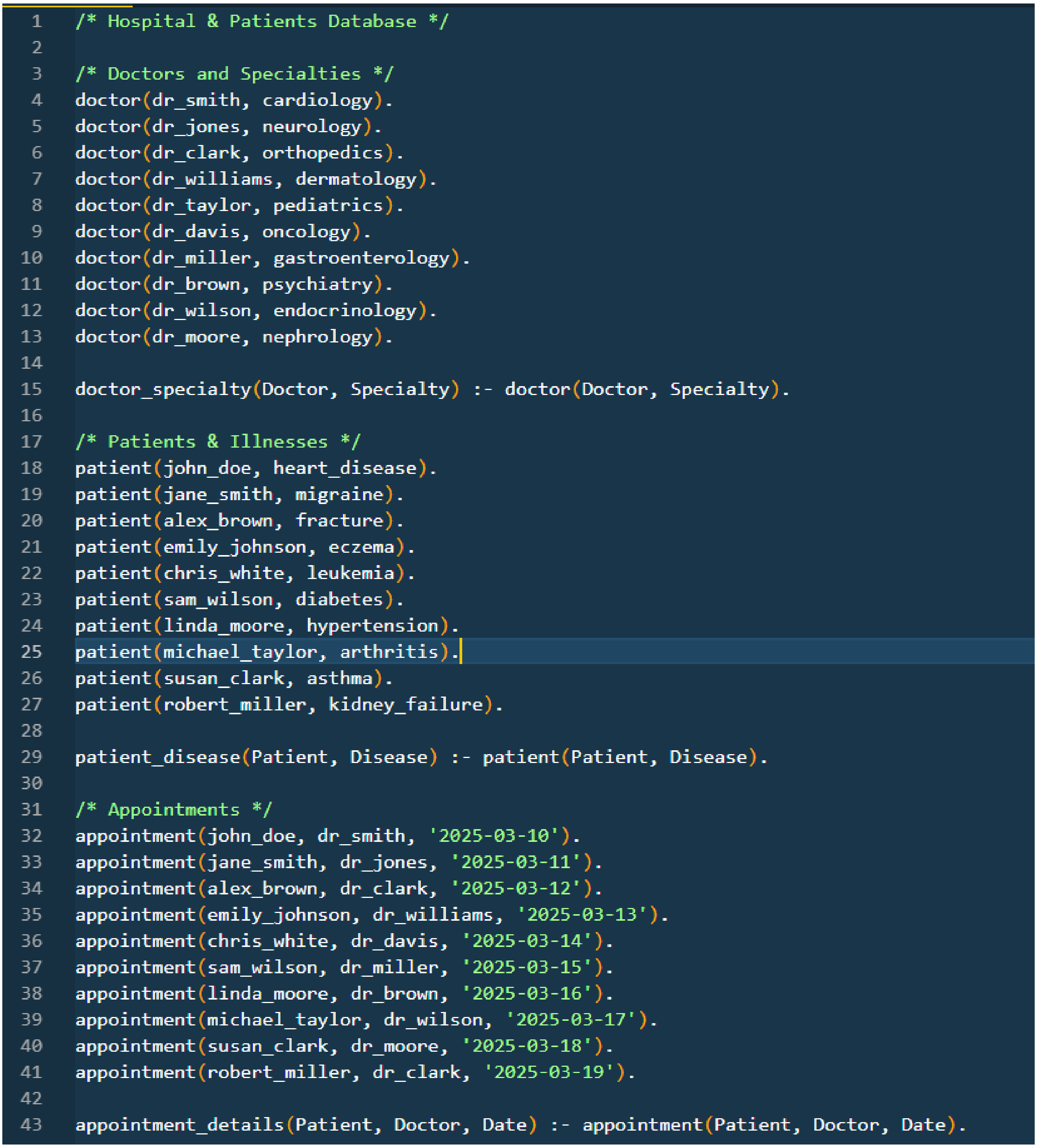
- **Programming Language**: Prolog
- **SWI-Prolog:** SWI-Prolog is an open-source Prolog interpreter that allows users to execute Prolog code, making it easier to develop, test, and debug logic-based programs.
- **Main Concepts**: Facts, Rules, Queries
- **Draw.io:** Used to design the Tree Figure and
- **Domain**: Hospital Management System

## iv. Diagram/Figure:

## Class Diagram:



## Sample Source Code pl file:



```prolog
/* Hospital & Patients Database */

/* Doctors and Specialties */
doctor(dr_smith, cardiology).
doctor(dr_jones, neurology).
doctor(dr_clark, orthopedics).
doctor(dr_williams, dermatology).
doctor(dr_taylor, pediatrics).
doctor(dr_davis, oncology).
doctor(dr_miller, gastroenterology).
doctor(dr_brown, psychiatry).
doctor(dr_wilson, endocrinology).
doctor(dr_moore, nephrology).

doctor_specialty(Doctor, Specialty) :- doctor(Doctor, Specialty).

/* Patients & Illnesses */
patient(john_doe, heart_disease).
patient(jane_smith, migraine).
patient(alex_brown, fracture).
patient(emily_johnson, eczema).
patient(chris_white, leukemia).
patient(sam_wilson, diabetes).
patient(linda_moore, hypertension).
patient(michael_taylor, arthritis).
patient(susan_clark, asthma).
patient(robert_miller, kidney_failure).

patient_disease(Patient, Disease) :- patient(Patient, Disease).

/* Appointments */
appointment(john_doe, dr_smith, '2025-03-10').
appointment(jane_smith, dr_jones, '2025-03-11').
appointment(alex_brown, dr_clark, '2025-03-12').
appointment(emily_johnson, dr_williams, '2025-03-13').
appointment(chris_white, dr_davis, '2025-03-14').
appointment(sam_wilson, dr_miller, '2025-03-15').
appointment(linda_moore, dr_brown, '2025-03-16').
appointment(michael_taylor, dr_wilson, '2025-03-17').
appointment(susan_clark, dr_moore, '2025-03-18').
appointment(robert_miller, dr_clark, '2025-03-19').

appointment_details(Patient, Doctor, Date) :- appointment(Patient, Doctor, Date).
```

## v. Sample Input/Output:

### Single Data:

### 1.Find the specialty of Dr. Smith:

**Input:** ?- doctor(dr_smith, Specialty).

**Output:**

```
Specialty = cardiology.
```

### 2.Find which disease John Doe has:

**Input:**?- patient(john_doe, Disease).

**Output:**

```
Disease = heart_disease.
```

### 3. Find the doctor treating Jane Smith:

**Input:**?- appointment(jane_smith, Doctor, Date).

**Output:**

```
Doctor = dr_jones,
Date = '2025-03-11'
```

### Filtering by Criteria:

### 4.Find all doctors specializing in neurology:

**Input:** ?- doctor(Doctor, neurology).

**Output:**

```
Doctor = dr_jones.
```

### 5. Find all patients who have diabetes:

**Input:**?- patient(Patient, diabetes).

**Output:**

```
Patient = sam_wilson
```

### 6. Find all patients who have appointments with Dr. Clark:

**Input:**?- appointment(Patient, dr_clark, Date).

**Output:**

```
Patient = alex_brown,
Date = '2025-03-12' .
```

### 7. Find the doctor treating Robert Miller and their specialty:

**Input:**?- appointment(robert_miller, Doctor, _), doctor(Doctor, Specialty).

**Output:**

```
 Doctor = dr_clark, Specialty = orthopedics.
```

### 8. Find all patients who have an appointment with a cardiologist:

**Input:**?- doctor(Doctor, cardiology), appointment(Patient, Doctor, Date).

**Output:**

```
Doctor = dr_smith,
Patient = john_doe,
Date = '2025-03-10'.
```

### 9. Find all doctors treating patients with heart disease:

**Input:**?- patient(Patient, heart_disease), appointment(Patient, Doctor, Date).

**Output:**

```
Patient = john_doe,
Doctor = dr_smith,
Date = '2025-03-10'.
```

### 10. Check if Dr. Wilson is an endocrinologist:

**Input:**?- doctor(dr_wilson, endocrinology).

**Output:**

```
True.
```

### 11. Check if Susan Clark has an appointment on '2025-03-18'

**Input:**?- appointment(susan_clark, Doctor, '2025-03-18').

**Output:**

```
Doctor = dr_moore.
```

### 12. List all doctors with appointments:

**Input:**?- `appointment(_, Doctor, _).`

**Output:**

```
Doctor = dr_smith ;
Doctor = dr_jones ;
Doctor = dr_clark ;
Doctor = dr_williams ;
Doctor = dr_davis ;
Doctor = dr_miller ;
Doctor = dr_brown ;
Doctor = dr_wilson ;
Doctor = dr_moore ;
Doctor = dr_clark.
```

### 13. List all unique patient diseases:

**Input:**`?- patient(_, Disease).`

**Output:**

```
Disease = heart_disease ;
Disease = migraine ;
Disease = fracture ;
Disease = eczema ;
Disease = leukemia ;
Disease = diabetes ;
Disease = hypertension ;
Disease = arthritis ;
Disease = asthma ;
Disease = kidney_failure.
```

### 14.Find all doctors and their specialties:

**Input:** `doctor(Doctor, Specialty).`

**Output:**

```
Doctor = dr_smith,
Specialty = cardiology ;
Doctor = dr_jones,
Specialty = neurology ;
Doctor = dr_clark,
Specialty = orthopedics ;
Doctor = dr_williams,
Specialty = dermatology ;
Doctor = dr_taylor,
Specialty = pediatrics ;
Doctor = dr_davis,
Specialty = oncology ;
Doctor = dr_miller,
Specialty = gastroenterology ;
Doctor = dr_brown,
Specialty = psychiatry ;
Doctor = dr_wilson,
Specialty = endocrinology ;
Doctor = dr_moore,
Specialty = nephrology.
```

## 15.Find all patients and their illnesses:

**Input:** patient(Patient, Disease).

**Output:**

```
Patient = john_doe,
Disease = heart_disease ;
Patient = jane_smith,
Disease = migraine ;
Patient = alex_brown,
Disease = fracture ;
Patient = emily_johnson,
Disease = eczema ;
Patient = chris_white,
Disease = leukemia ;
Patient = sam_wilson,
Disease = diabetes ;
Patient = linda_moore,
Disease = hypertension ;
Patient = michael_taylor,
Disease = arthritis ;
Patient = susan_clark,
Disease = asthma ;
Patient = robert_miller,
Disease = kidney_failure
```

## 16.Find all appointments:

**Input:**?- appointment(Patient, Doctor, Date).

**Output:**

```
Patient = john_doe,
Doctor = dr_smith,
Date = '2025-03-10' ;
Patient = jane_smith,
Doctor = dr_jones,
Date = '2025-03-11' ;
Patient = alex_brown,
Doctor = dr_clark,
Date = '2025-03-12' ;
Patient = emily_johnson,
Doctor = dr_williams,
Date = '2025-03-13' ;
Patient = chris_white,
Doctor = dr_davis,
Date = '2025-03-14' ;
Patient = sam_wilson,
Doctor = dr_miller,
Date = '2025-03-15' ;
Patient = linda_moore,
Doctor = dr_brown,
Date = '2025-03-16' ;
Patient = michael_taylor,
Doctor = dr_wilson,
Date = '2025-03-17' ;
Patient = susan_clark,
Doctor = dr_moore,
Date = '2025-03-18' ;
Patient = robert_miller,
Doctor = dr_clark,
Date = '2025-03-19'.
```

## vi. Conclusion and Challenges:

**Conclusion:** This Prolog-based hospital management system efficiently organizes and manages hospital data. It allows for tracking various entities, such as doctors, patients, appointments, and surgeries, in a clear and structured manner. The system also provides tools for financial management, including billing and staff salary calculations.

**Challenges:**

1. **Scalability**: As the database grows, the performance of the Prolog system might decline due to the complexity of querying and handling large datasets.
2. **Data Maintenance**: Prolog's lack of an interactive user interface could make data entry and updates cumbersome without integration with other systems or a front-end interface.
3. **Advanced Features**: The system lacks more complex features like patient history, medical prescriptions, or integration with external databases for comprehensive healthcare management.