VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

Database Systems - CO2013

Assignment 1 Report - Group 6

Hospital Management System

Advisor(s): Vo Thi Kim Anh

Student(s): Tran Dang Hien Long ID 2252449
Tran Dinh Dang Khoa ID 2211649
Tran Phu Duc ID 2210814
Nguyen Nhat Huy ID 2053042

HO CHI MINH CITY, APRIL 2025

Contents

Overview

The Hospital Management System (HMS) is a comprehensive solution designed to streamline hospital operations, including patient care, staff management, billing, and diagnostics. The system centralizes data management, automates workflows, and ensures secure access to medical records, billing, and diagnostic information.

Key Objectives:

- Improve operational efficiency by digitizing patient records, staff assignments, and billing.
- Enhance patient care through structured medical history tracking, diagnostic test management, and surgery records.
- Facilitate seamless communication between departments (e.g., assigning doctors/nurses to patients, equipment maintenance).
- Ensure compliance with healthcare data security standards.

Value Proposition

1 Why Users Want This System

The Hospital Management System appeals to different user groups:

- **Patients:** Want easy access to health records, appointments, and bills
- **Doctors & Nurses:** Need efficient access to patient data and streamlined workflows
- **Administrators:** Require tools for resource management and simplified billing
- **Technicians:** Need systems for equipment maintenance tracking

2 User Benefits

HMS provides key benefits:

- **Patients:** Better care coordination and clearer billing
- **Doctors & Nurses:** Less paperwork and faster access to patient data
- **Administrators:** Improved efficiency and better resource allocation
- **Technicians:** Organized maintenance workflows

System Requirements

3 Functional Requirements

3.1 User Access Control

- Role-based access control for Admins, Doctors, Nurses, Patients, and Technicians.

3.2 Patient Management

- Add, update, delete patient records (demographics, contact info, medications).
- Track medical history, allergies, and insurance details.

3.3 Staff Management

- Admin can add/update/delete Doctors, Nurses, and Technicians.
- Assign Doctors/Nurses to patients via `AssignDoctor` and `AssignNurse` relationships.

3.4 Billing & Insurance

- Generate bills with initial, covered, and final amounts.
- Link insurance policies to patients and calculate coverage.

3.5 Diagnostics & Surgery

- Record diagnostic tests (e.g., blood tests, X-rays) and associate them with equipment.
- Track surgeries, outcomes, and equipment used.

3.6 Equipment Maintenance

- Technicians log maintenance activities for medical equipment.

3.7 Reporting

- View patient lists, recent surgeries, and diagnostic results based on roles.

4 Non-Functional Requirements

4.1 Security

- Secure access to sensitive patient and hospital data.
- Mechanisms to ensure data integrity and confidentiality.

4.2 Performance

- Efficient data retrieval and processing.
- Responsive system interactions for users.

4.3 Scalability

- Designed with a modular architecture to allow for future expansion and addition of features.
- Ability to handle increasing amounts of data and users over time.

4.4 Usability

- Intuitive user interface with role-specific dashboards.
- Clear error messages and input validations.

4.5 Compliance

- Adherence to relevant healthcare data privacy standards (e.g., HIPAA-like principles).

Database Design

5 List of strong entities

A robust hospital database must capture essential information about key elements like patients and doctors. These core components are represented as strong entities, with relationships describing their interactions. This chapter focuses on the attributes, relationships, and significance of each strong entity.

5.1 Patient Record

The Patient Record entity stores essential personal and medical information for each patient:

- Record Number (Primary Key) - Unique patient identifier
- First Name - Patient's first name
- Last Name - Patient's last name
- Gender - Patient's gender
- Contact Information - Phone number and email
- Home Address - Residential address (Street, District, City)
- Date of Birth - Patient's birth date
- Current Medications - Present prescriptions
- Emergency Contact - Emergency contact's phone number

5.2 Employee

The Employee entity captures essential information for each staff member to facilitate HR management and operational efficiency:

- Employee ID (Primary Key) - Unique staff identifier (e.g., D001 for Doctor)
- Name - Employee's full name
- Gender - Employee's gender
- Date of Birth - Employee's birth date
- Job Type - Employment category (Doctor, Nurse, Technician, etc.)
- Experience - Years in profession
- Salary - Annual or hourly compensation
- Contact Details - Phone number(s) and email
- Start Date - Employment commencement date
- Department Number (Foreign Key) - Department identifier

Employee serves as a superclass with specialized subclasses:

1. Doctor:
    - Specialty: Medical field (Cardiology, Pediatrics, etc.)
    - Certificate: Professional qualifications
2. Nurse:
    - Specialty: Medical focus (Diagnosis, Nursing)
3. Technician:
    - Specialty: Technical focus (Electricity, Medical Facility, etc.)
    - Responsibility: Specific duties related to specialty
4. Other:
    - Job-type: Position (Receptionist, Janitor, etc.)
    - Responsibility: Specific duties related to position

5.3 Department

The Department entity represents specialized medical service units:

- Department ID (Primary Key) - Unique department identifier
- Department Name - Name (e.g., Cardiology, Pediatrics)
- Location - Physical location within or outside hospital

5.4 Room

The Room entity represents hospital spaces serving various purposes:

- Room Number (Primary Key) - Unique room identifier
- Department Number (Foreign Key) - Associated department
- Room Type - Purpose classification (Patient care, ICU, etc.)
- Room Name - Descriptive name
- Status - Current room condition

5.5 Equipment

The Equipment entity tracks medical tools inventory:

- Equipment ID (Primary Key) - Unique equipment identifier
- Name - Equipment name (MRI Machine, Surgical Scalpel, etc.)
- Type - Category (Medical, Diagnostic, Surgery)
- Status - Availability status

5.6 Surgery

The Surgery entity records surgical procedure data:

- Surgery ID (Primary Key) - Unique surgery identifier
- Type of Surgery - Procedure classification
- Date of Surgery - Operation date

- Outcome - Procedure result (Successful, Complications)
- Complications - Any issues encountered

5.7 Diagnostic Test

The Diagnostic Test entity tracks medical examinations:

- Test ID (Primary Key) - Unique test identifier
- Test Name - Examination name (Blood Test, X-Ray, etc.)
- Description - Brief test description
- Date - Test date
- Results - Test outcomes

5.8 Insurance

The Insurance entity manages patient coverage information:

- Insurance ID (Primary Key) - Unique insurance identifier
- Policy Number - Insurance policy number
- Priority - Multiple policy application sequence
- Provider - Insurance company name
- Status - Policy status (Active, Pending, Expired)
- Coverage Percentage - Percentage covered
- Coverage Limit - Maximum coverage amount

5.9 Billing

The Billing entity tracks patient care costs:

- Billing ID (Primary Key) - Unique billing identifier
- Date Issued - Creation date

- Initial Amount - Pre-insurance charge
- Cover Amount - Insurance-covered portion
- Final Amount - Patient responsibility
- Due Date - Payment deadline
- Status - Payment status

5.10 Payment

The Payment Record entity tracks billing transactions:

- Payment ID (Primary Key) - Unique payment identifier
- Payment Date - Transaction date
- Method - Payment method (Cash, Credit Card)
- Amount Paid - Transaction amount
- Payment Receipt Number - Transaction reference
- Notes - Additional information

6 Relationships between strong entity types

6.1 Patient - Billing

Links patients to their billing records:

- **Denoted by:** Have
- **Relationship's type:** 1:N (one patient to many billings)
- **Implementation:** PatientID added to Billing as Foreign Key

6.2 Patient - Insurance

Links patients to their insurance policies:

- **Denoted by:** Have
- **Relationship's type:** 1:N (one patient to many insurance policies)
- **Implementation:** PatientID added to Insurance as Foreign Key

6.3 Insurance - Billing

Links billing records to covering insurance policies:

- **Denoted by:** Cover
- **Relationship's type:** N:M (many-to-many)
- **Implementation:** Junction table with BillingID and InsuranceID

6.4 Payment - Billing

Links billing records to payments:

- **Denoted by:** Pay
- **Relationship's type:** 1:N (one billing to many payments)
- **Implementation:** BillingID added to Payment as Foreign Key

6.5 Doctor - Patient and Nurse - Patient

Links healthcare providers to patients:

- **Denoted by:** Assign
- **Relationship's type:** 1:N (one doctor/nurse to many patients)
- **Implementation:** Two junction tables (Assign_Doc, Assign_Nurse)

6.6 Technician - Equipment

Links technicians to maintained equipment:

- **Denoted by:** Maintains
- **Relationship's type:** 1:N (one technician to many equipment pieces)
- **Implementation:** New table with Equipment ID, TechID, Type, and Date

6.7 Surgery - Doctor - Patient

Links surgeries to performing doctors and treated patients:

- **Denoted by:** Perform_Surgery
- **Relationship's type:** Ternary relationship
- **Implementation:** Junction table with SurgeryID, DoctorID, and PatientID

6.8 Surgery - Equipment

Links surgeries to used equipment:

- **Denoted by:** Use_In_Surgery
- **Relationship's type:** N:M (many-to-many)
- **Implementation:** Junction table with SurgeryID and EquipID

6.9 Diagnostic Test - Nurse - Patient

Links tests to performing nurses and tested patients:

- **Denoted by:** Perform_Test
- **Relationship's type:** Ternary relationship
- **Implementation:** Junction table with TestID, NurseID, and PatientID

6.10 Diagnostic Test - Equipment

Links tests to utilized equipment:

- **Denoted by:** Use_In_Test
- **Relationship's type:** N:M (many-to-many)
- **Implementation:** Junction table with TestID and EquipID

6.11 Employee - Department

Links employees to their departments:

- **Denoted by:** Work_in
- **Relationship's type:** 1:N (one department to many employees)
- **Implementation:** Department_Number added to Employee as Foreign Key

6.12 Manager - Department

Links managers to managed departments:

- **Denoted by:** Manage
- **Relationship's type:** 1:1 (one manager to one department)
- **Implementation:** Manager_ID added to Department as Foreign Key

6.13 Department - Room

Links departments to contained rooms:

- **Denoted by:** Contains
- **Relationship's type:** 1:N (one department to many rooms)
- **Implementation:** DepartmentID added to Room as Foreign Key

7 Weak entities

7.1 Allergies

Dependent on Patient entity, with composite primary key:

- PatientID (Foreign Key)
- Allergy (specific patient allergy)

7.2 Medical History

Dependent on Patient entity, with composite primary key:

- PatientID (Foreign Key)
- Type (medical issue category)
- Description (issue details)
- Treatment (treatment details)
- Stage (issue severity)

7.3 Bed

Dependent on Room entity, representing individual patient beds.

8 Key attributes of entities and their description

8.1 Patient record

- **Key attribute:** PatientID
- **Description:** Unique identifier enabling accurate tracking of patient medical information

8.2 Employee

- **Key attribute:** EmployeeID
- **Description:** Unique identifier for proper management of staff roles and responsibilities

8.3 Department

- **Key attribute:** DepartmentID
- **Description:** Unique identifier for organizational management of hospital units

8.4 Room

- **Key attribute:** RoomID
- **Description:** Unique identifier for facility space management

8.5 Equipment

- **Key attribute:** EquipmentID
- **Description:** Unique identifier for tracking and managing medical devices

8.6 Surgery

- **Key attribute:** SurgeryID
- **Description:** Unique identifier for procedure documentation and tracking

8.7 Diagnostic Test

- **Key attribute:** TestID
- **Description:** Unique identifier for medical test tracking

8.8 Insurance

- **Key attribute:** InsuranceID
- **Description:** Unique identifier for managing different patient coverage plans

8.9 Billing

- **Key attribute:** BillingID
- **Description:** Unique identifier for tracking financial transactions

8.10 Payment

- **Key attribute:** ReceiptID
- **Description:** Unique identifier for tracking payment transactions

8.11 Allergies

- **Key attribute:** Allergy (partial key)
- **Description:** Descriptive label that, with PatientID, uniquely identifies patient allergies

8.12 Medical History

- **Key attribute:** Type (partial key)
- **Description:** Medical issue category that, with PatientID, uniquely identifies history entries

9 Identify constraint

9.1 Uniqueness constraint

Each entity has a primary key providing unique identification among entity instances, as listed in the previous section.

9.2 Cardinality & participation constraint

9.2.1 Patient - Billing

- **Cardinality constraint**: 1:N (one patient to many billings)
- **Participation constraint**: Billing has total participation; Patient has partial participation

9.2.2 Patient - Insurance

- **Cardinality constraint**: 1:N (one patient to many insurances)
- **Participation constraint**: No total participation on either side

9.2.3 Insurance - Billing

- **Cardinality constraint**: M:N (many insurances to many billings)
- **Participation constraint**: No total participation on either side

9.2.4 Payment - Billing

- **Cardinality constraint**: 1:N (one billing to many payments)
- **Participation constraint**: Only Payment has total participation

9.2.5 Doctor - Patient

- **Cardinality constraint**: 1:N (one doctor to many patients)
- **Participation constraint**: No total participation on either side

9.2.6 Nurse - Patient

- **Cardinality constraint**: 1:N (one nurse to many patients)
- **Participation constraint**: Only Patient has total participation

9.2.7 Technician - Equipment

- **Cardinality constraint**: M:N (many technicians to many equipment)
- **Participation constraint**: No total participation on either side

9.2.8 Surgery - Doctor - Patient

- **Cardinality constraint**: M:N:1 (many doctors to many surgeries to one patient)

9.2.9 Surgery - Equipment

- **Cardinality constraint**: M:N (many surgeries to many equipment)
- **Participation constraint**: Only Surgery has total participation

9.2.10 Diagnostic Test - Nurse - Patient

- **Cardinality constraint**: M:N:1 (many nurses to many tests to one patient)

9.2.11 Diagnostic Test - Equipment

- **Cardinality constraint**: M:N (many tests to many equipment)
- **Participation constraint**: Only Diagnostic Test has total participation

9.2.12 Employee - Department

- **Cardinality constraint**: 1:N (one department to many employees)
- **Participation constraint**: Both sides have total participation

9.2.13 Manager - Department

- **Cardinality constraint**: 1:1 (one manager to one department)
- **Participation constraint**: Only Department has total participation

9.2.14 Room - Department

- **Cardinality constraint**: 1:N (one department to many rooms)
- **Participation constraint**: Only Department has total participation

9.2.15 Room - Bed (Identity relationship)

- **Cardinality constraint**: 1:N (one room to many beds)
- **Participation constraint**: Only Bed has total participation

9.2.16 Patient - Allergies (Identity relationship)

- **Cardinality constraint**: 1:N (one patient to many allergies)
- **Participation constraint**: Only Allergies has total participation

9.2.17 Patient - Medical History (Identity relationship)

- **Cardinality constraint**: 1:N (one patient to many history entries)
- **Participation constraint**: Only Medical History has total participation

10 Entity-relationship diagram (ERD)

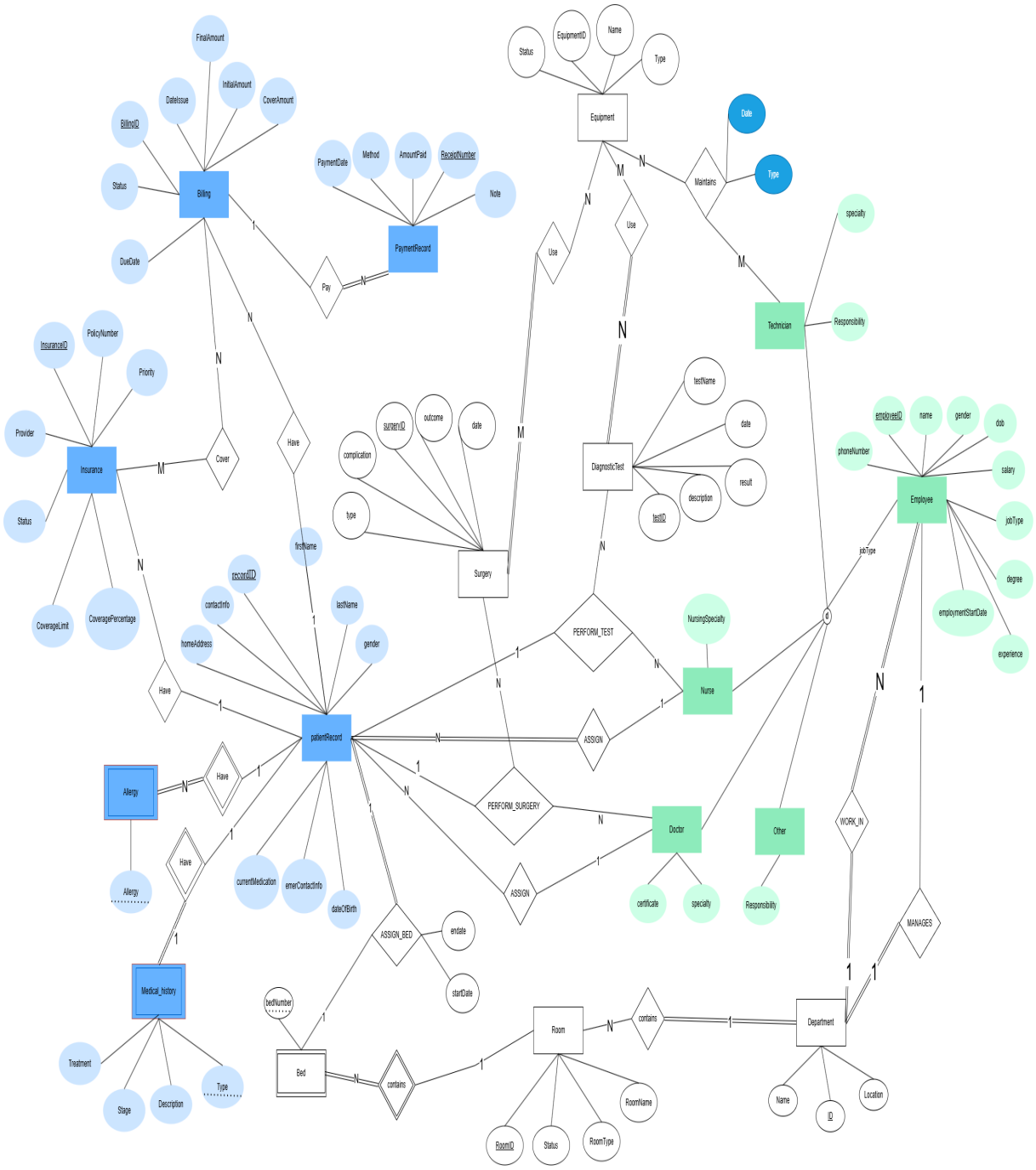

Figure 10.1: ERD for the hospital database

11 Mapping to relational schema & specifying constraint

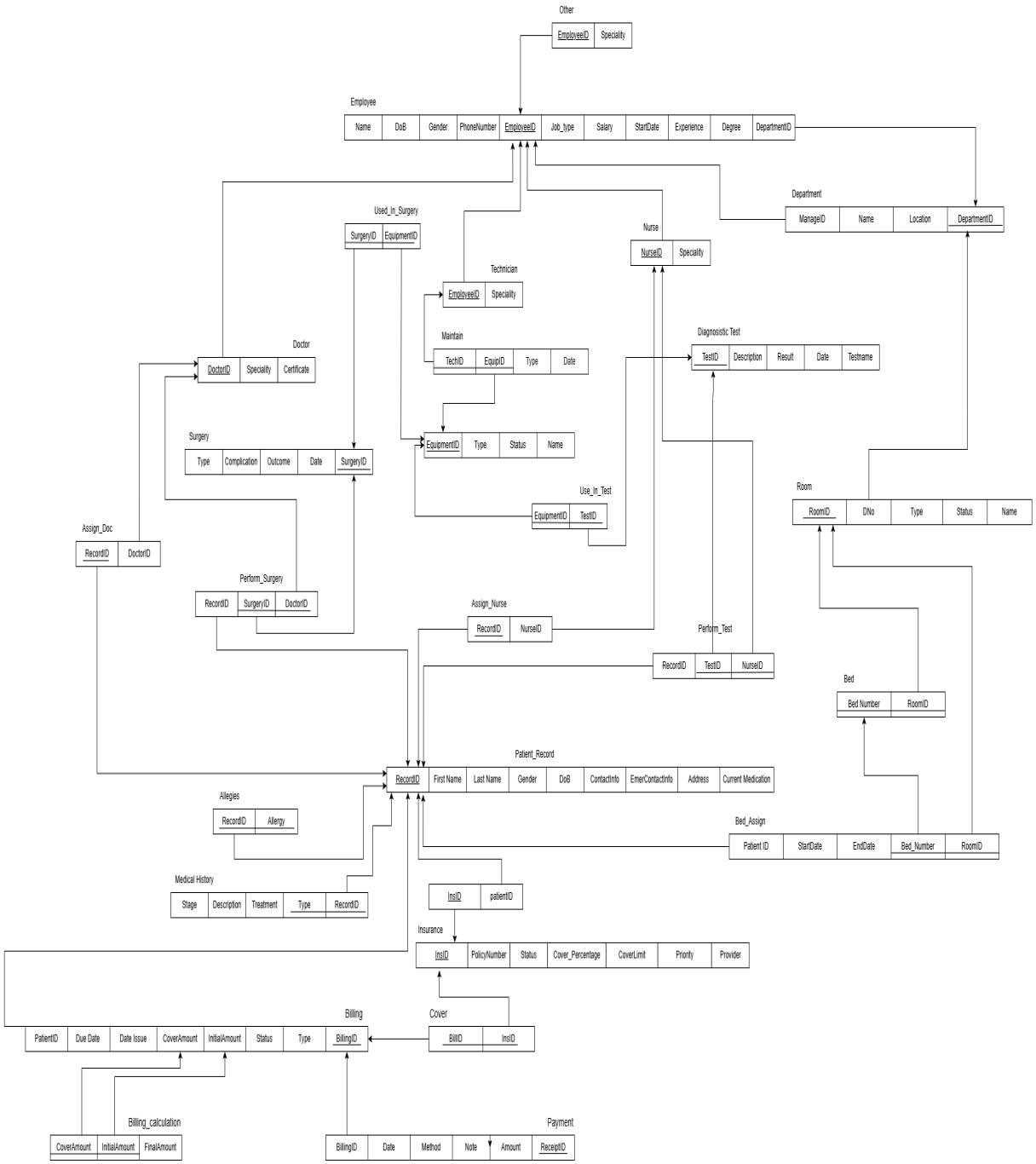

Figure 11.1: Relational schema for the hospital database