HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
Faculty of Computer Science and Engineering

Database Systems

Group 1 - CC02

Assignment 1

HOSPITAL DATABASE SYSTEM

Supervisor: Dr. Phan Trong Nhan

Students:

Name	ID
Do Thanh Trung	2252853
Tran Gia Linh	2252434
Huynh Nguyen Ngoc Anh	2252022
Nguyen Trinh Tien Dat	2252147
Huynh Thanh Duy	2252114

Ho Chi Minh City, November 2023

Contents

Name	Contrib	Assigned Tasks
Do Thanh Trung	100%	EERD Design + Mapping 1. Introduction & 4. Discussion Report Integration
Tran Gia Linh	100%	EERD Design + Mapping 2.2 Relationships
Huynh Nguyen Ngoc Anh	100%	EERD Design + Mapping 2.3 Constraints
Nguyen Trinh Tien Dat	100%	EERD Design + Mapping 3. EER-to-Relational Mapping
Huynh Thanh Duy	100%	EERD Design + Mapping 2.1 Strong and Weak entities

Table 1: Member contribution

Section 1

Introduction

1.1 Context

This hospital database project aims to design a structured system to manage key entities within a hospital. The goal is to facilitate efficient management of different entities: storing, retrieving, and organizing large amounts of data while maintaining accuracy and integrity.

This database design must also reflect the relationships between various actors (e.g., employees and patients) and resources (e.g., departments, medication). By using Enhanced Entity Relationship (EER) model and transforming it into a relational schema, the database must maintain consistency and scalability.

By analyzing the database specifications, we aim to break down two primary elements of the (E)ER model: the distinct entities within the system and the relationships between them.

1.2 Entities and Relationships

1.2.1 Strong Entities

- Employees:
  - Doctors: professionals who diagnose, treat, and examine patients.
  - Nurses: assist doctors and care for patients.
- Speciality: an area of medical expertise, such as cardiology or neurology.

- Departments: divisions within the hospital, such as Rehab or HR.
- Patients:
  - Inpatients: admitted to the hospital for at least one night for treatment.
  - Outpatients: visit the hospital for medical care without staying overnight.
- Health Care Service:
  - Treatment: provided to patients to cure or manage illnesses.
  - Examination: performed to diagnose a patient's condition.
- Medication: a type of medications prescribed to treat patients.
- Prescription: specify medication and dosage, issued by doctors.
- Providers: supply medication to the hospital.
- Batch: contains medication supplied by provider.

1.2.2 Weak Entity

There are no weak entities in our database design.

1.2.3 Relationships

Relationships between entities are also crucial for the design. Firstly, we begin by identifying **binary relationships**:

No	Relationship	Description
1	`works_for`	An employee works for a department
2	`is_specialized_in`	An employee is specialized in some specialities.
3	`manages`	A dean, who is a doctor, manages a department.
4	`care`	Nurse takes care of inpatients

5	`has`	A treatment or examination has a medication (or prescription).
6	`consists_of`	A prescription consists of some types of medications with some quantity.
7	`imported_from`	A medication type is imported from batches.
8	`supplies`	A provider supply some batches of medication.

Table 1.1: All binary relationships in the hospital database

To represent the relationship between patients, doctors and the medication that the doctors provide, we introduce 2 **ternary relationships**:

No	**Relationship**	**Description**
1	`treats`	An inpatient can receive treatments from at least one doctor.
2	`examines`	An outpatients can have many examinations with their examining doctor.

Table 1.2: All ternary relationships in the hospital database.

To mitigate complex relationships that can make the database harder to manage, understand, and query effectively, the design avoids relationships that involve four or more entities. Thus, we focus entirely on binary and ternary relationships.

Section 2

Enhanced Entity Relationship Diagram

The Introduction section enables us to develop an EER diagram to capture key entities and relationships. It is the foundation of our database design.

Overview

The Enhanced Entity-Relationship Diagram (EERD) is a detailed extension of the traditional ER model. It provides a richer framework for complex database designs.

This section covers key concepts such as **strong** and **weak entities** and the **relationships** that connect entities and the various **constraints** that ensure the integrity and validity of the data. These constraints include:

- specialization constraints, which define subclass relationships
- semantic constraints that preserve logical data meanings.
- domain constraints that restrict attribute values.
- constraints on NULL values that enforce data completeness and accuracy

To begin with, We would like to present the EER diagram that the our team design:

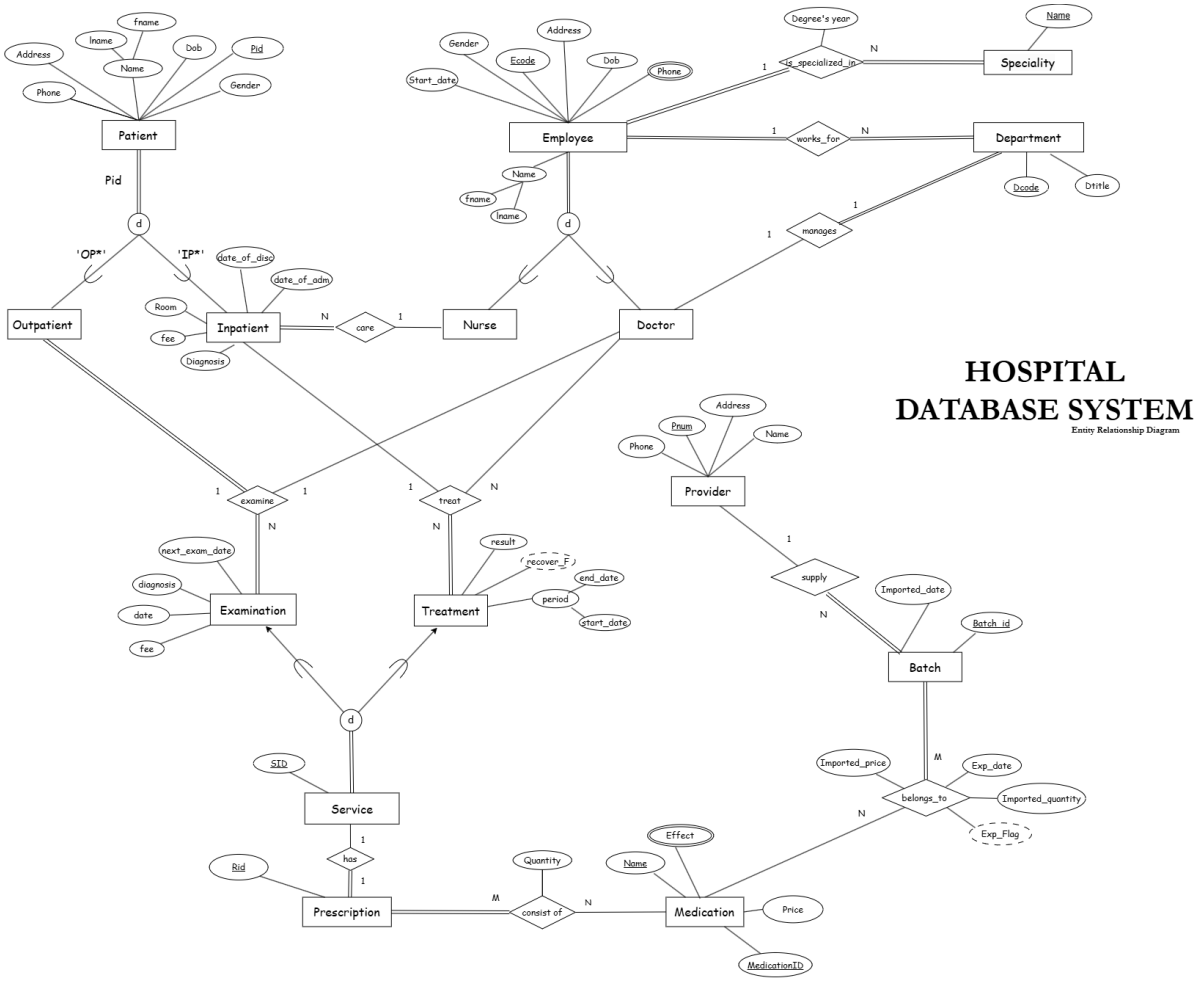

Figure 2.1: EERD design of the hospital database

In the following sections, we will demonstrate how the EER diagram can be drawn from various pieces of information.

2.1 Strong and Weak entities

Entity name	**Attribute**	**Description**
Employee	Ecode	Primary key: Employee's code
	fname	Single-valued: First name
	lname	Single-valued: Last name
	Address	Single-valued
	Gender	Single-valued
	Dob	Single-valued: Date of birth
	Start_date	Single-valued
	Phone	Multi-valued
Doctor (subclass of Employee)	-	-
Nurse (subclass of Employee)	-	-
Department	Dcode	Primary key: Department's code
	Dtitle	Single-valued: Department's title
Patient	Pid	Primary key: Patient ID
	fname	Single-valued: First name
	name	Single-valued: Last name
	Address	Single-valued
	Gender	Single-valued
	Dob	Single-valued: Date of birth
	Phone	Single-valued

Inpatient (subclass of Patient)	date_of_disc	Single-valued: Date of discharge
	date_of_adm	Single-valued: Date of admission
	fee	Single-valued
	room	Single-valued
	diagnosis	Single-valued
Outpatient (subclass of Patient)	-	-
Service	SID	Primary key: Service ID
Examination (subclass of Service)	fee	Single-valued: Examination fee
	diagnosis	Single-valued
	date	Single-valued: Examination date
	next_exam_date	Single-valued: Next examination date
Treatment (subclass of Service)	result	Single-valued: Treatment result
	start_date	Single-valued: Start date
	end_date	Single-valued: End date
	recover_F	Single-valued: Recovery flag
Prescription	Rid	Primary key: Reception ID
Medication	ID	Primary key
	Name	Single-valued
	Price	Single-valued
	Effect	Multi-valued
Batch	Batch_id	Primary key
	Imported_date	Single-valued

Provider	Pnum Name Address Phone	Primary key: Provider's number Single-valued Single-valued Single-valued
Speciality	Name	Primary Key

2.2 Relationships

*** Notation** Partial: —
Total: =

a. Binary relationships

- 1:1 relationship: `Manages`, `Has`.
- 1:N relationship: `Works_for`, `Care`, `Supplies`, `Is_specialized_in`.
- M:N relationship: `Consists_of`, `Imported_from`.

Manages

Dean, who is a doctor, manages a department.

Associated entities	Doctor, Department
Attributes	None
Cardinality	1:1 **Each department has one dean.** An employee must belong to a specific department and only a doctor can be a dean (managing a department). Therefore, **a doctor can only manage one department.**
Membership	Doctor: — Department: = **Not every** doctor is a dean. Every department **must** have a dean.

Has

A healthcare service (examination, treatment) that involves a medication prescription.

Associated entities	Service, Prescription
Attributes	None
Cardinality	1:1 Each treatment or examination has a prescription. A prescription belongs to a treatment or examination.
Membership	Service: — Prescription: = Whenever performing a healthcare service, the doctor(s) **need not** to give any medication to the patient. A prescription must belong to a treatment or an examination.

Works_for

Employee works for a department.

Associated entities	Employee, Department
Attributes	None
Cardinality	N:1 Each department can have **many** employees. An employee works for a **specific** department.
Membership	Employee: = Department: = An employee **must** work for a department. A department has **at least one** employee working for.

Care

Nurse takes care of inpatient.

Associated entities	Nurse, Inpatient
Attributes	None
Cardinality	1:N An inpatient is taken care of by **a** nurse. A nurse can take care of **many** patients at the same time.
Membership	Nurse: — Inpatient: = **Each** inpatient is taken care of by a nurse. A nurse **can** take care of many patients at the same time

Supplies

Provider supplies a batch of medication to the hospital.

Associated entities	Provider , Batch
Attributes	None
Cardinality	1:N A batch of medication belongs to the supply of **one** provider. A provider can supply **many** batches of medication to the hospital.
Membership	Provider : — Batch: = A provider **may have not** supplied medication to the hospital yet. A batch of medications **must** be provided by a provider.

Is_specialized_in

Employee is specialized in a speciality.

Associated entities	Employee , Speciality
Attributes	Degree 's year: The year that the employee is qualified for the speciality.
Cardinality	1:N A particular speciality degree belongs to one employee. An employee can be specialized in many specialties.
Membership	Employee : = Speciality: - It is assumed that either a nurse or a doctor must be specialized in **at least one** speciality. A speciality can be qualified by none working in the hospital.

Consists_of

A medical prescription consists of medication.

Associated entities	Prescription, Medication
Attributes	**Quantity**: amount of medication in the prescription.
Cardinality	M:N A medication can be in **many** prescriptions. A prescription can have **many** types of medications in it. This is because each treatment or examination can have medication(s), and a treatment/examination associated with a prescription.
Membership	Prescription : = Medication: - A medical prescription must have medication(s) in it. A medication needs not to be mentioned in any prescription.

Imported_from

A type of medication is imported from a batch of medication.

Associated entities	Medication, Batch
Attributes	**Imported_price**: The price at which the medication was imported in the batch. **Imported_quantity**: The amount of a type of medication that was imported from the medication batch. **Exp_date**: Expired date of a type of medication which is imported from the batch. **Exp_flag**: If medication from a batch is expired, the flag is 1 and is 0 otherwise.
Cardinality	M:N A medication can be in **many** prescriptions. A prescription can have **many** types of medications in it. This is because each treatment or examination can have medication(s), and a treatment/examination associated with a prescription.
Membership	Medication: = Batch: = Medication is provided by a provider (unit is batch). A batch must consist of medication(s).

b. Ternary relationships

There are 2 ternary relationships in the EERD: Treat and Examines.

Treat

Doctor(s) give treatment to an inpatient.

Entities	Inpatient, Treatment, Doctor
Attributes	None
Cardinality	1:N:N A doctor and a treatment has one inpatient. An inpatient can receive a treatment from many doctors. An inpatient and a doctor can have many treatments.
Membership	Inpatient: — Treatment: = Doctor: — An inpatient **may not** have any treatment yet when they are admitted to the hospital. A treatment **always occurs** when doctors treat inpatients. Sometimes, a doctor **has no** inpatient to treat.

Examines

A doctor examines an outpatient.

Entities	Outpatient, Doctor, Examination
Attributes	None
Cardinality	1:1:N A particular examination of one doctor corresponding to one outpatient. An examination of one outpatient is conducted by a doctor. The outpatients can have many examinations with their examining doctor.
Membership	Outpatient: — Examination: = Doctor: — An outpatient **may not** have any examination yet. A doctor may give **no** examination to any outpatient. An examination **occurs** when a doctor examines an outpatient.

2.3 Constraints

2.3.1 Constraints of specialization

Patient

- Inpatient and Outpatient are predicate-defined subclasses of Patient: we specify the condition of membership in the Inpatient and Outpatient subclass by the condition that their Pid must begin with “OP” and “IP” respectively.
- Inpatient and Outpatient subclass must be disjoint and total subsets of Patient: A patient must be classified as either a inpatient or an outpatient, but not both. Employee superclass:

Employee

- Nurse and Doctor are user-defined subclasses of Employee.
- Nurse and Doctor subclass must be disjoint and total subsets of Patient: an employee must be classified as either a nurse or a doctor, but not both.
- Nurse and Doctor subclass must be disjoint and total subsets of Patient: an employee must be classified as either a nurse or a doctor, but not both.

Service

- Examination and Treatment are user-defined subclasses of Service.
- Examination and Treatment subclass must be disjoint and total subsets of Patient: In the context of the hospital here, a service that hospital offers to patients must only be either a Treatment or an Examination.

2.3.2 Semantic constraints

- Unique code for patients must follow the format “OP” or “IP” followed by 9 digits, specifically “OP” for outpatients and “IP” for inpatient.

- A doctor who manages a department must have a speciality related to that field and must have been awarded that speciality degree for more than 5 years.
- Medications out-of-date are automatically marked in the database.
- When a inpatient is recovered, his or her last treatment must be recorded and marked as "Recovered".
- Discharge date of a patient can only be filled once their last treatment is confirmed as "Recovered".
- All `End_date` values in Treatment must greater than or equal to their corresponding `Start_date` value for each record.

2.3.3 Domain constraints

- All the dates and dob must have a valid format "YYYY-MM-DD", ensuring that the month is between 1 and 12 and the day is between 1 and 31.
- All the Degree's year values of Speciality must have "YYYY" format, with Y is a digit
- All the phone number must be in valid format "+84" followed by 9 digits (for the context that the hospital is in Vietnam), except the phone values of Provider.
- All of the `Price` values of the `Medication` entity, `imported_price` and `imported_quantity` of `belongs_to` relationship, `quantity` values of `consists_of` relationship and `fee` values of `Inpatient` entity must be greater than 0 and is measured in VND.
- Gender values for Employee and Patient must be either "M" or "F".

2.3.4 Constraint on NULLs

Entity type	Non-NULL Attribute (primary key excluded)
Employee	- Fname - Lname - Start_date - Dno
Department	- Dtitle - Mgr_code
Inpatient	- Date_of_asm - Nurse_code
Patient	- Fname - Lname - Phone
Medication	- Name
Belongs_to	- Imported_quantity - Imported_price - Exp_date
Batch	- Provider_num - Imported_date
Provider	- Name - Phone
Treatment	- Start_date
Examination	- Date

Section 3

EER-to-Relational Mapping

In this section, we map the EERD to a relational schema, transforming the conceptual into a logical design.

There are nine mandatory steps to perform mapping from an EER diagram to a relational schema. We follow each of these steps to construct the relational schema of the hospital database. They are:

- **ER-**
  - ☐ Step 1: Mapping of Regular Entity Types
  - ☐ Step 2: Mapping of Weak Entity Types
  - ☐ Step 3: Mapping of Binary 1:1 Relationship Types
  - ☐ Step 4: Mapping of Binary 1:N Relationship Types
  - ☐ Step 5: Mapping of Binary M:N Relationship Types
  - ☐ Step 6: Mapping of Multivalued Attributes
  - ☐ Step 7: Mapping of N-ary Relationship Types
- **EER-**
  - ☐ Step 8: Mapping Specialization or Generalization
  - ☐ Step 9: Mapping of Union Types (Categories)

3.1 Mapping of Strong Entity Types

In this step, each strong entity type in the EER model is mapped to a relation.

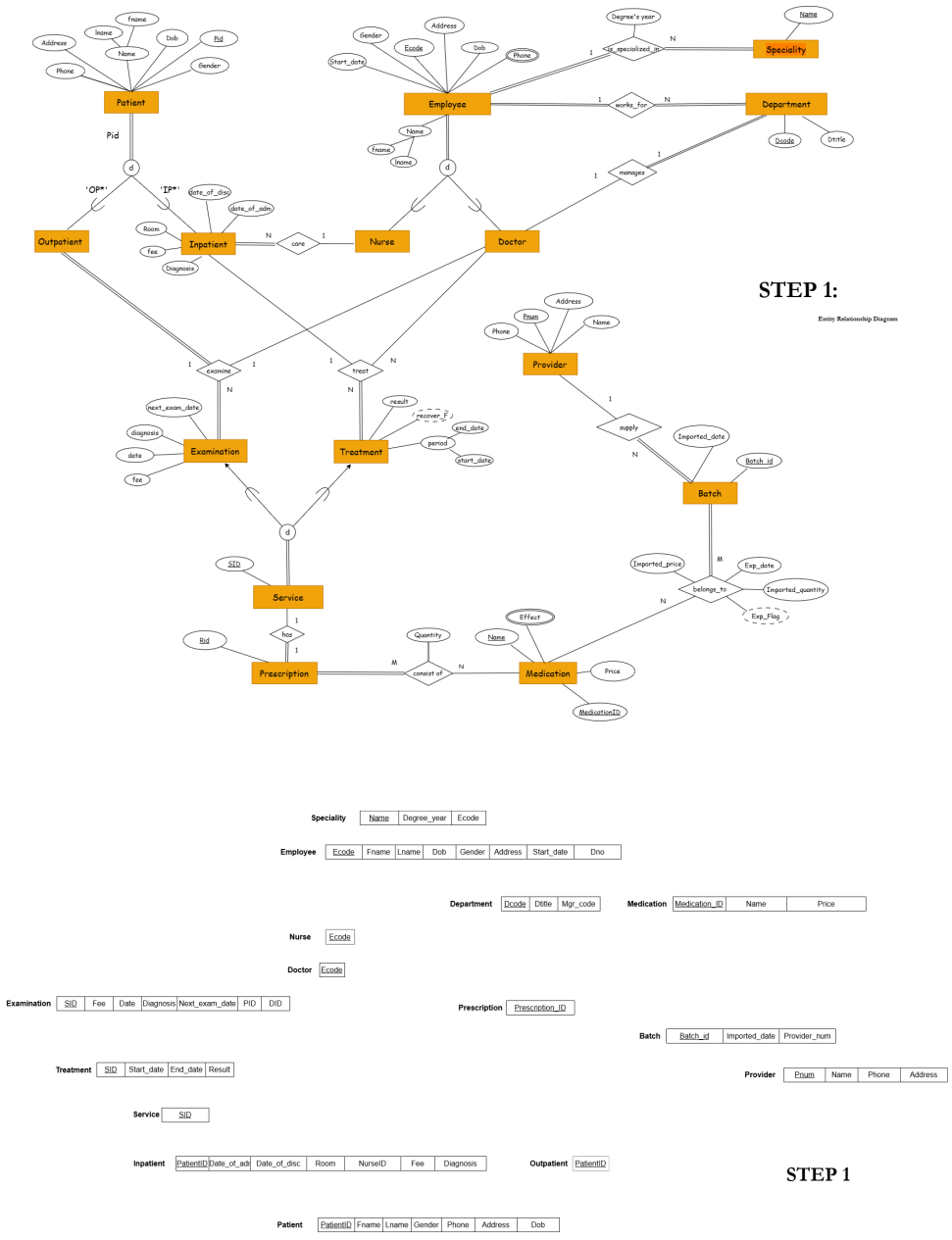

Figure 3.1: Mapping of Strong Entity Types

3.2 Mapping of Weak Entity Types

There are none of weak entity types in our database so we omit this step.

3.3 Mapping of Binary 1:1 Relationship Types

For binary 1:1 relationships, we can add the primary key of one entity as a foreign key of the other.

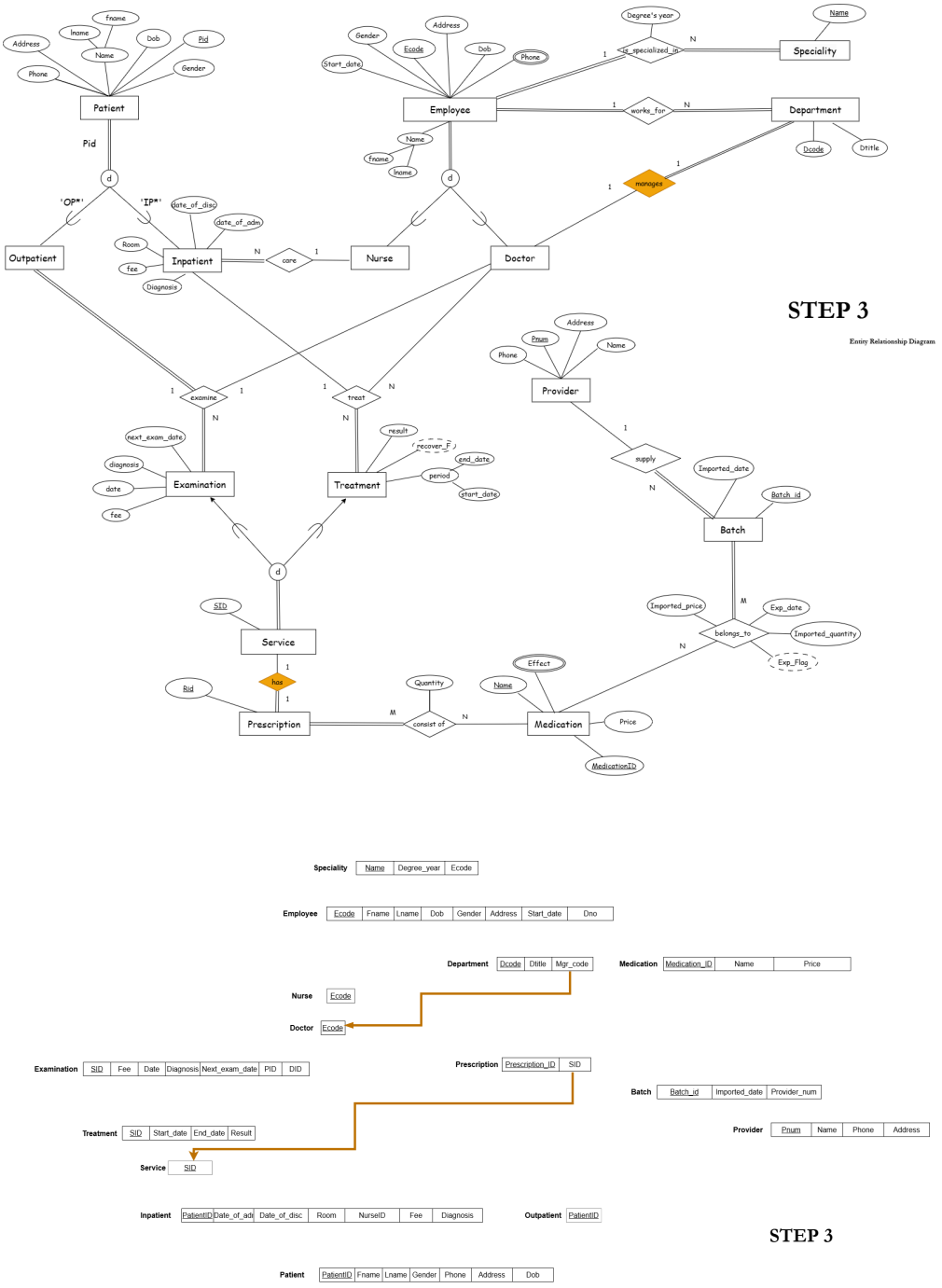

Figure 3.2: Mapping of Binary 1:1 Relationship Types

3.4 Mapping of Binary 1:N Relationship Types

In a 1:N binary relationship, we include the primary key of the ”1” side entity as a foreign key in the relation representing the ”N” side.

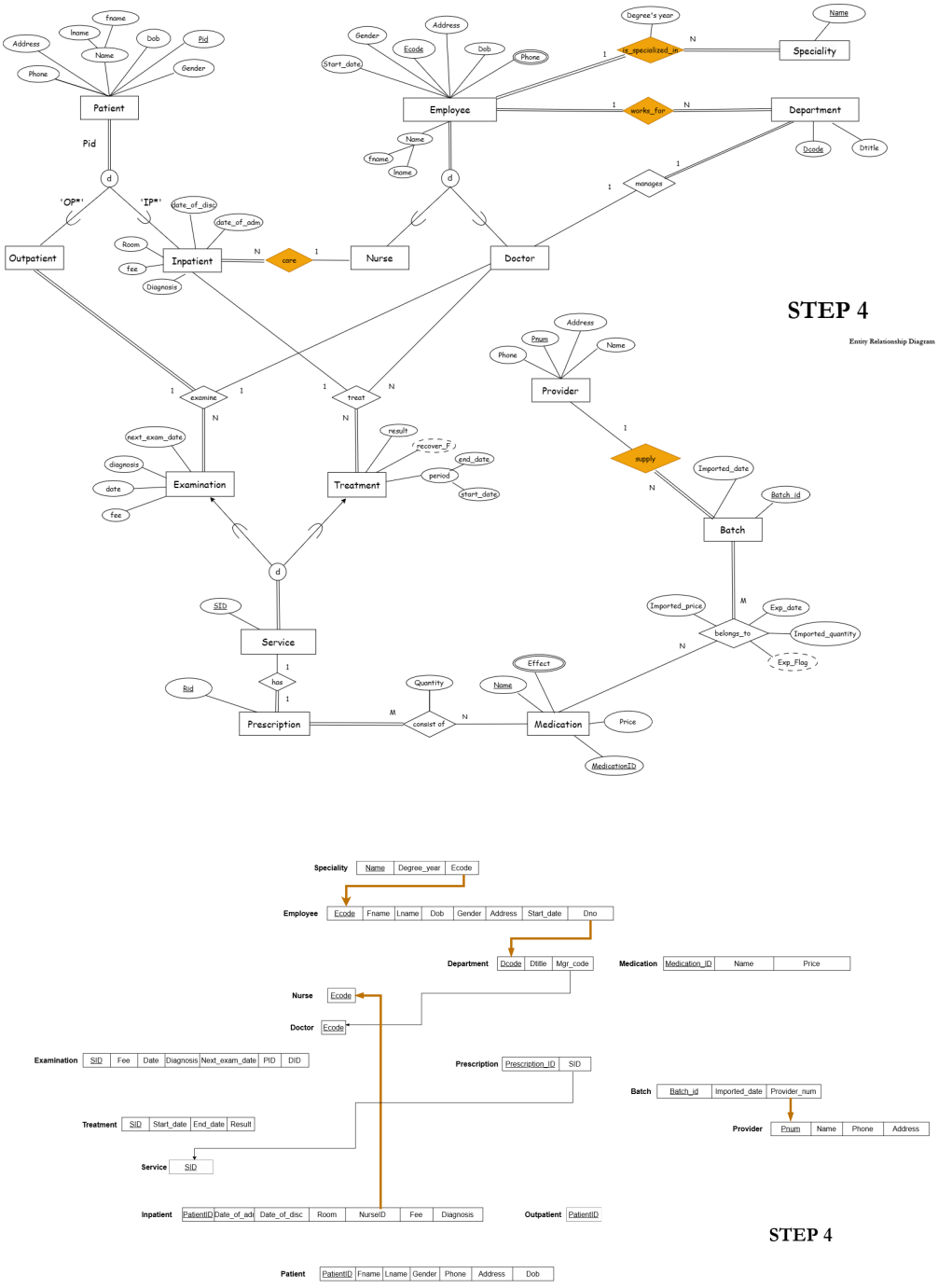

Figure 3.3: Mapping of Binary 1:N Relationship Types

3.5 Mapping of Binary M:N Relationship Types

For M:N relationships, we create a new relation (table) to represent the relationship. The primary keys of both participating entities become foreign keys in the new relation.

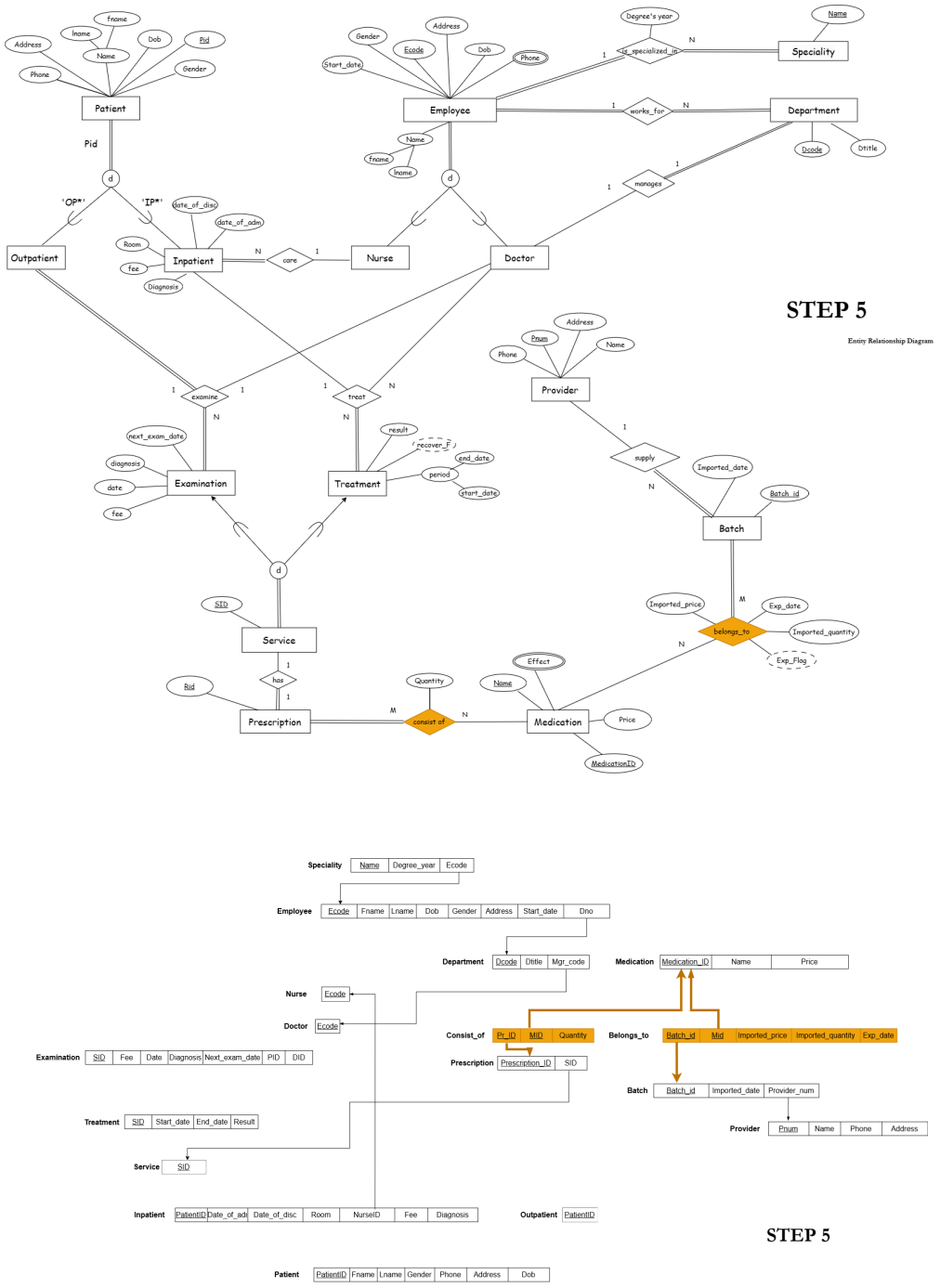

Figure 3.4: Mapping of Binary M:N Relationship Types

3.6 Mapping of Multivalued Attributes

When an entity has multivalued attributes, we create a separate relation for each multivalued attribute.

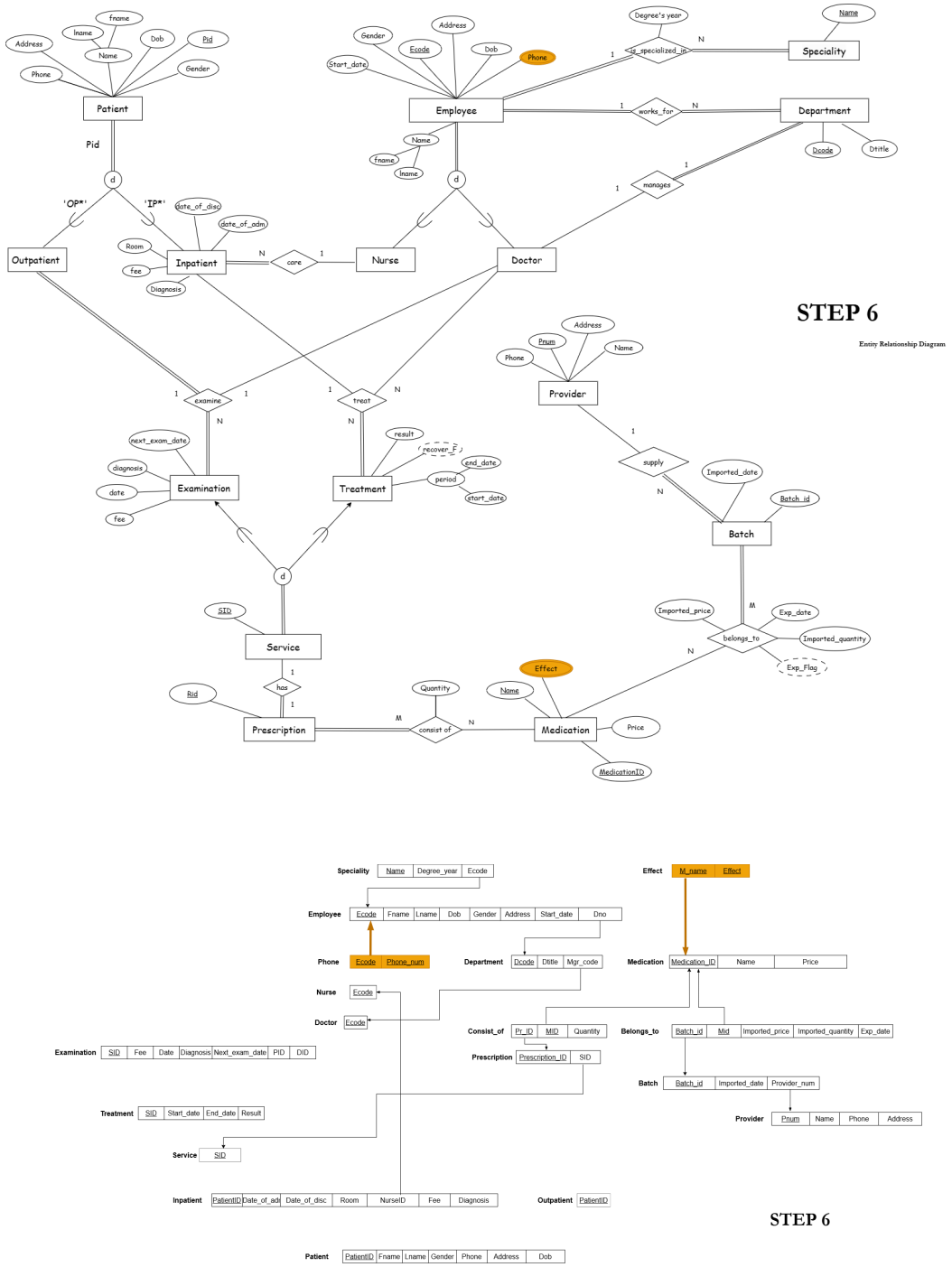

Figure 3.5: Mapping of Multivalued Attributes

3.7 Mapping of N-ary Relationship Types

For N-ary relationships, we map them to a new relation. The primary keys of all participating entities are included as foreign keys in this new relation.

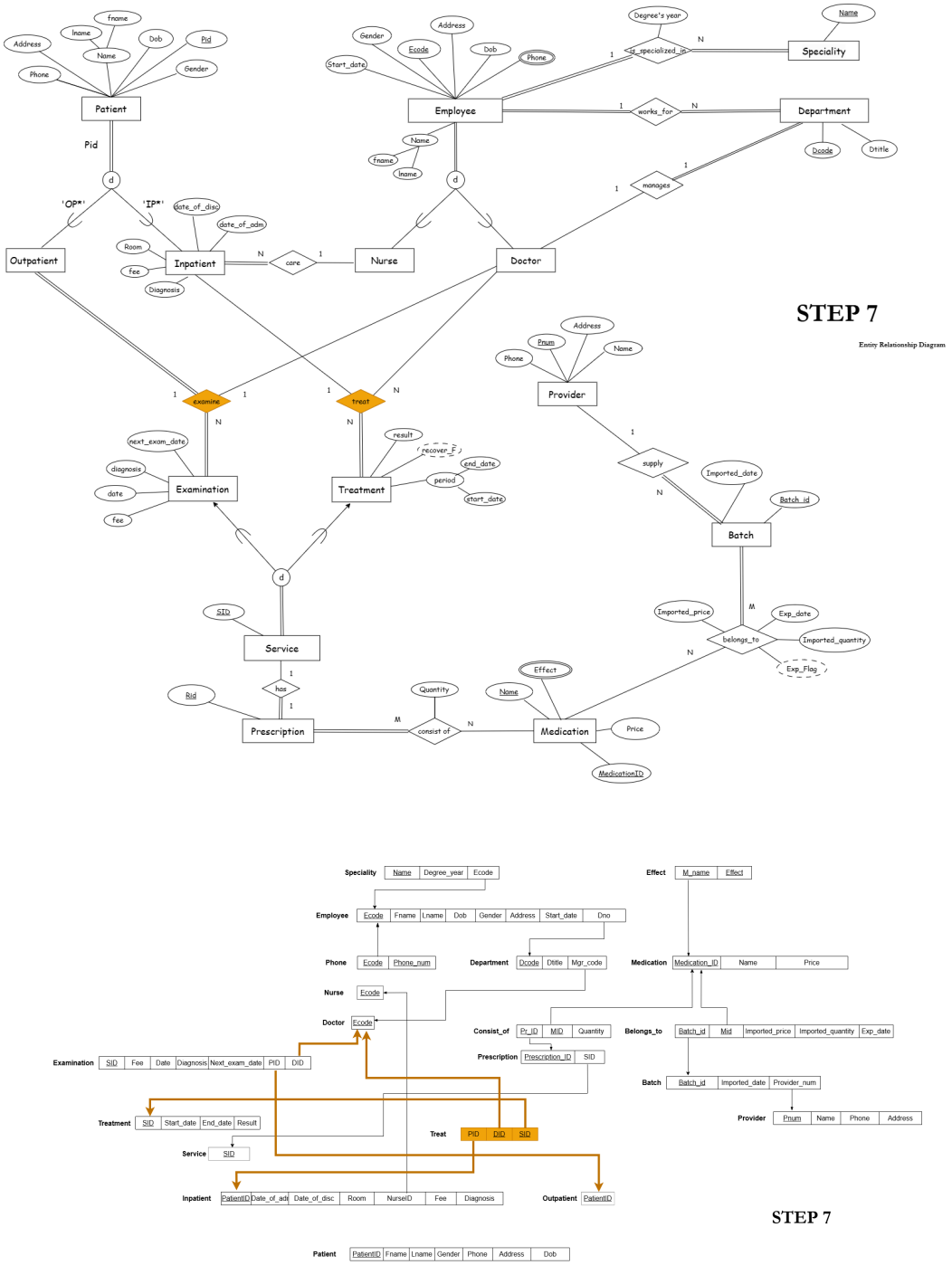

Figure 3.6: Mapping of N-ary Relationship Types

3.8 Mapping Specialization or Generalization

Specialization or generalization can be mapped using one of several strategies. Here we choose option 8A for all of our mapping.

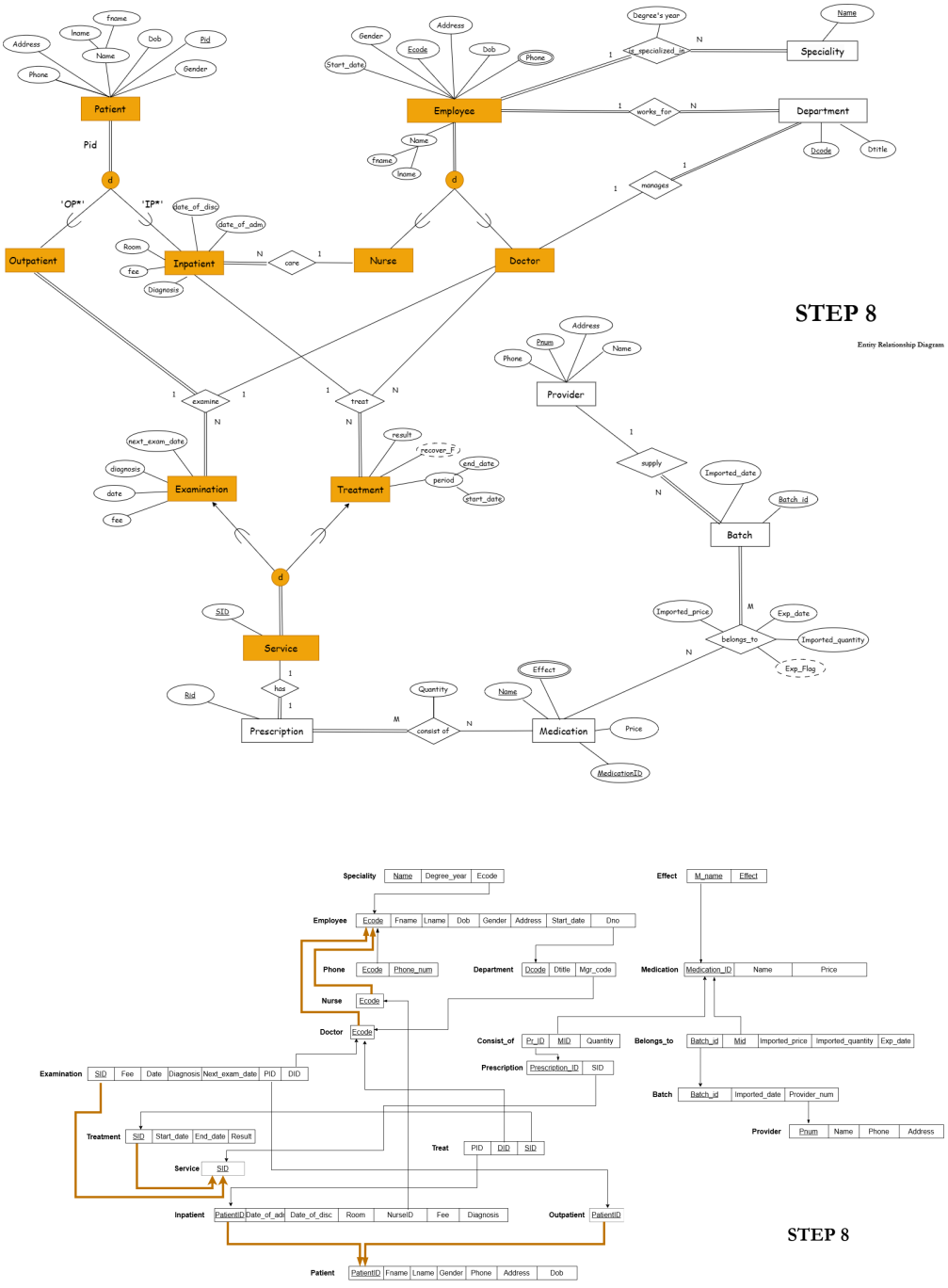

Figure 3.7: Mapping Specialization or Generalization

3.9 Mapping of Union Types (Categories)

There are none of Union Types in our EERD.

3.10 Relational Schema

Finally, we present the final relational schema resulting from the above mapping stages:

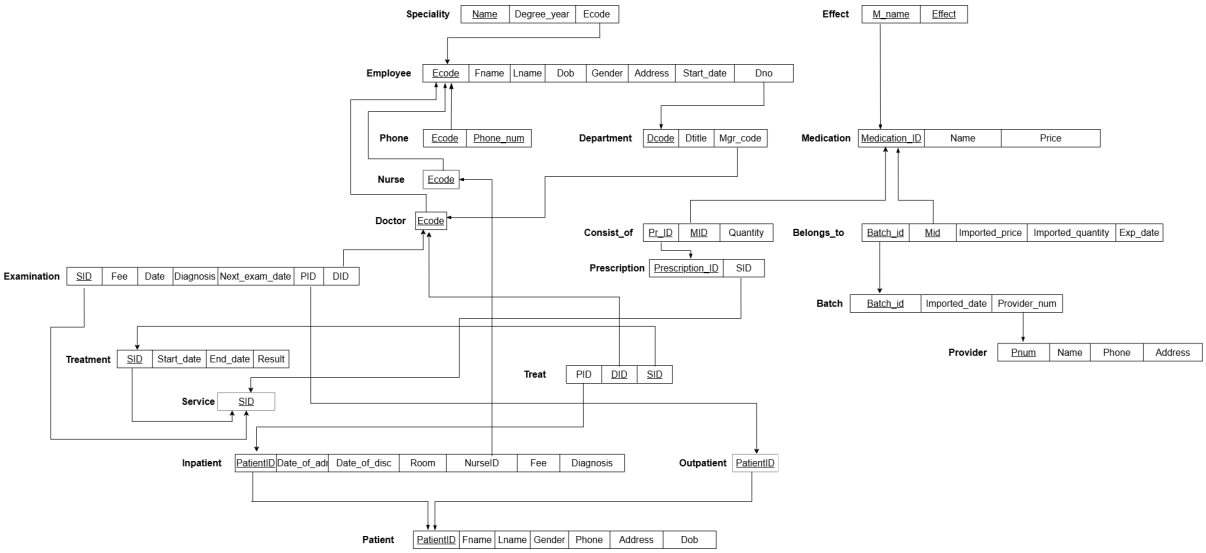

Figure 3.8: Relational Schema for the hospital database.

Section 4

Discussion

This final section reflects on the database design and discusses challenges, improvements, and future extensions that may enhance the system.

Managing Information and Workflow

Designing a hospital database involves managing large volumes of information, such as patient records, staff details, and medical supplies. Ensuring that entities and relationships are clearly defined is crucial for effective workflows.

Data Integrity

Maintaining data integrity is essential, especially when dealing with sensitive patient information. This database ensures that records are securely stored and easily accessible.

Scalability

Scalability is also important, as the hospital database must handle increasing data volumes without slowing down performance. The translation of the EER model into a relational schema ensures efficient querying and data storage.

Avoiding Complexity

The decision to avoid relationships with more than three entities prevents unnecessary complexity. The system remains easy to maintain and flexible.

Security and Privacy

Security and privacy are key concerns, with measures like role-based access control and encryption implemented to safeguard sensitive data.

Section 5

Diagrams Source

EER Diagram: Google Drive

Relational Schema: Google Drive