



Politecnico  
di Torino

DBG  
MG

# Database design

---

Example

# Conceptual design (1/2)

---

- Database design stages
- Design example: problem specifica
- Design example: main concepts
- Design example: model refinement (I)
- Design example: model refinement (II)
- Design example: model refinement (III)

# Conceptual Design (2/2)

---

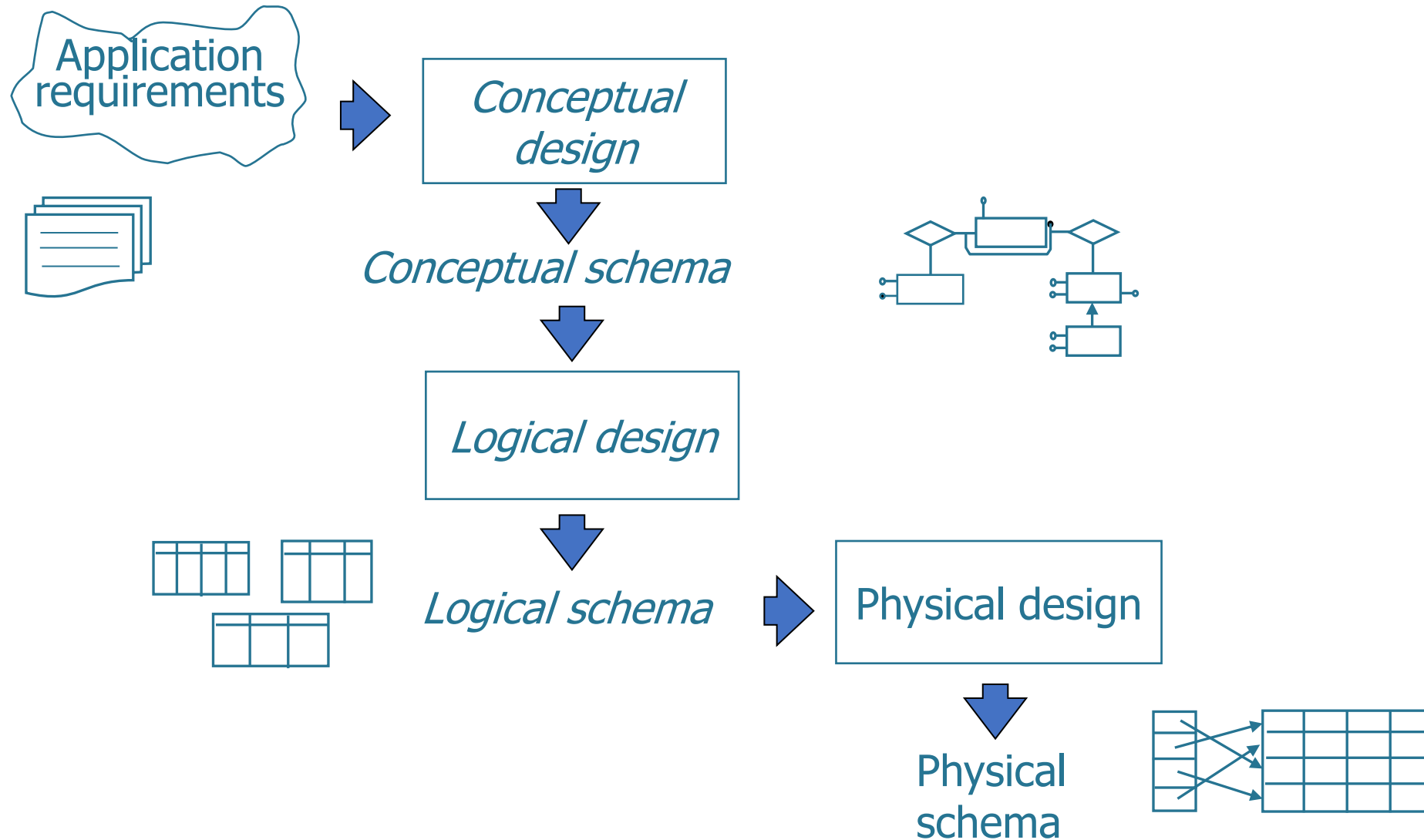
- Design example: representation of time (I)
- Design example: representation of time (II)
- Design example: representation of time (III)

# Database design

---

Database design stages

# Database design stages



# Requirements collection and analysis

---

- Requirements collection
  - problems that the application must solve
  - static and dynamic features of the application
- Requirements analysis
  - clarification and organization of the specifications
- Interconnected and scarcely standardizable activities

# Requirements sources

---

- Application users
  - interviews
  - written documentation
- Existing documentation
  - regulations
  - internal rules
  - forms
- Existing systems
  - applications to replace or interact with

# Requirements collection

---

- System users play an important role
  - high-level users have a more general view, but they often do not know the details
  - different users can provide different information (complementary or contradictory)



# Requirements collection

---

- In practice:
  - verify that the collected information is consistent and correct
  - verify also with examples (related to general and borderline cases)
  - require precise definitions and classifications
  - distinguish the essential requirements from nice-to-have and less stringent requirements
  - proceed through subsequent refinements

# Requirements analysis

---

- In practice:
  - choose the right abstraction level
  - standardize the structure of the sentences
  - avoid convoluted sentences
  - identify synonyms/homonyms and unify the terms
  - make explicit all references between terms
  - build a glossary of terms

# Conceptual design

---

- Various project strategies have been proposed
- The most effective strategy is hybrid
  - the basic concepts are identified (the most relevant entities and relationships)
  - the initial project is progressively refined by adding the attributes, the cardinality of the relationships, the hierarchies, other entities and relationships
- If the problem is highly complex, it can be broken down into subproblems to solve separately and integrate later

# Conceptual design: general criteria

---

- If a concept has significant properties or describes classes of objects with autonomous existence
  - entity
- If a concept has a simple structure and has no relevant properties
  - attribute (possibly multivalued)
- If two or more concepts are related
  - relation
- If a concept is a particular case of another one
  - hierarchy

# Quality of a conceptual scheme

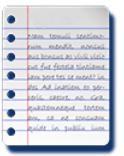
---

- Correctness
  - use of appropriate model constructs
  - absence of syntactic and semantic errors
- Completeness
  - representation of all the concepts of interest
- Minimality
  - every requirement is represented only once in the schema
  - all redundancies are verified and documented
- Legibility

# Database design

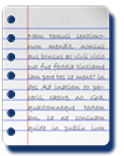
---

Problem requirements



# Problem requirements

- We want to represent a database for the management of a medical examination booking system within a Local Health Authority (ASL), considering the following information.
- Each patient is characterized by the health card number, name, surname, address, date of birth, place of birth and age.
- ASL hospitals are characterized by a numerical code, name and address.

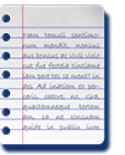


# Problem requirements

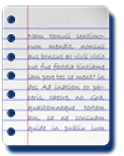
- Each hospital is divided into departments identified by a unique numeric code within the hospital and characterized by department name and telephone number.
- The department staff is identified by a Social Security Number (SSN). Name, surname and address are also known.
- Among the staff, for doctors the list of specializations achieved is known and for the volunteers the name of the association they belong to (if available) is stored.



# Problem requirements

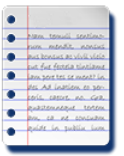


- The medical examinations that can be performed are characterized by a numerical code and a textual description (e.g., X-ray exam, etc.).
- For specialistic examinations, the doctor who carries out the visit and a description of the diet to follow (if necessary) are also stored.
- The laboratories that perform the examinations are identified by a unique code within the hospital and they are characterized by the name of the laboratory, the location plan and the room number.



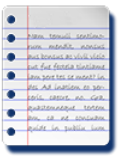
# Problem requirements

- For each member of the laboratory staff, the days and laboratories in which he/she works are stored. Pay attention to the fact that during the same day each staff member can work in more than one laboratory.



# Problem requirements

- Each exam requires a reservation. For each exam reservation made by a patient, the date and time of the exam, the laboratory where it is performed, the cost of the ticket and the information about the exam being urgent or not are stored.
- Please note that each patient can make multiple reservations for the same exam on different dates and the same exam cannot be repeated on the same day by the same patient, even in different laboratories.



# Problem requirements

- Each doctor can take on different roles during his/her career (e.g. assistant, head physician, etc.). We want to keep track of the roles each doctor has taken on during his/her career and the related time periods (start date, end date).
- Keep in mind that each doctor cannot take on more than one role at the same time, but he/she can take on the same role in different time periods.

# Database design

---

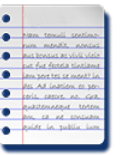
Main concepts

# Main concepts identification

---

- Text analysis to identify the most important concepts
  - the main entities of the ER diagram
  - any links between entities

# Patient Concept



- Each *patient* is characterized by health card number, name, surname, address, date of birth, place of birth and age.

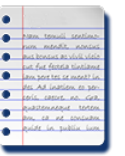
# Patient Concept

---

**Patient**



# Hospital concept



- ASL *hospitals* are characterized by a numeric code, a name and an address.

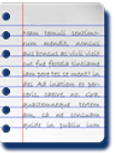
# Hospital concept

---

**Patient**

**Hospital**

# Department concept



- Each hospital is divided into *departments* identified by a unique numeric code within the hospital and characterized by the department name and the telephone number.

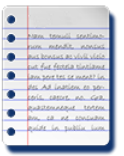
# Department concept

---

**Patient**

**Hospital**

**Department**



# Staff concept

- The *department staff* is identified by a Social Security Number (SSN).
- Name, surname and address are also known.
- Among the staff, for **doctors** the list of specializations achieved is known and for the **volunteers** the name of the association they belong to (if available) is stored.

# Staff concept

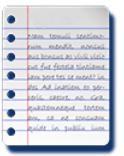
---

**Patient**

**Staff**

**Hospital**

**Department**

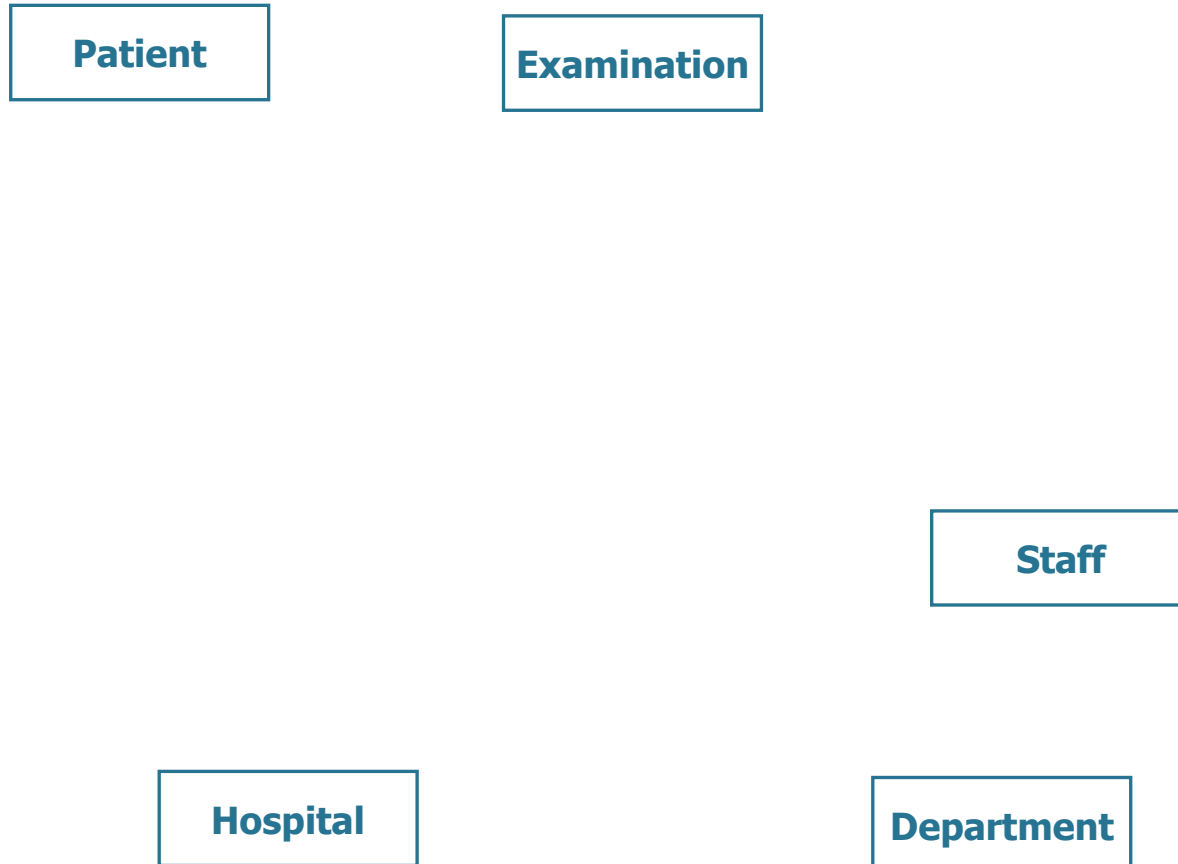


# Examination concept

- The *medical examinations* that can be performed are characterized by a numerical code and a textual description (e.g. X-ray exam, etc.).
- For specialist examinations, the doctor who carries out the visit and a description of the diet to follow (if necessary) are also stored.

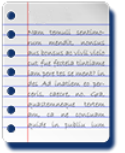
# Examination concept

---





# Laboratory concept



- The *laboratories* that perform the tests are identified by a unique code within the hospital and are characterized by the name of the laboratory, the location plan and the room number.

# Laboratory concept

---

**Patient**

**Examination**

**Laboratory**

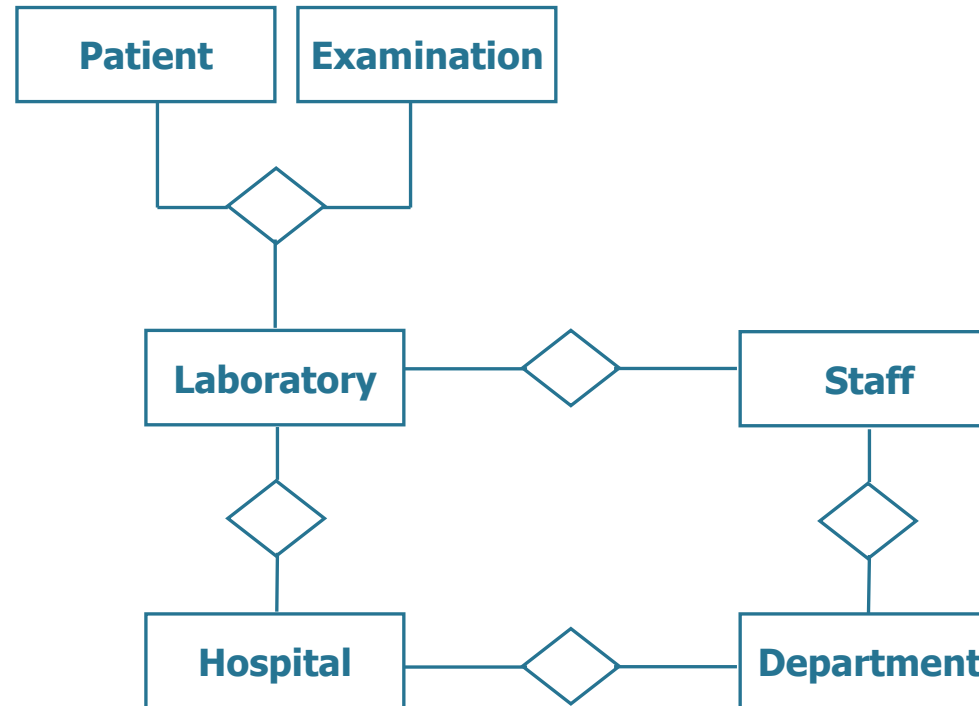
**Staff**

**Hospital**

**Department**

# Main concepts

- Main concepts
  - patient
  - examination
  - laboratory
  - hospital
  - department
  - staff



# Database design

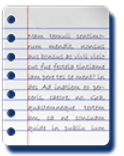
---

## Model refinement (I)

# Concepts refinement

---

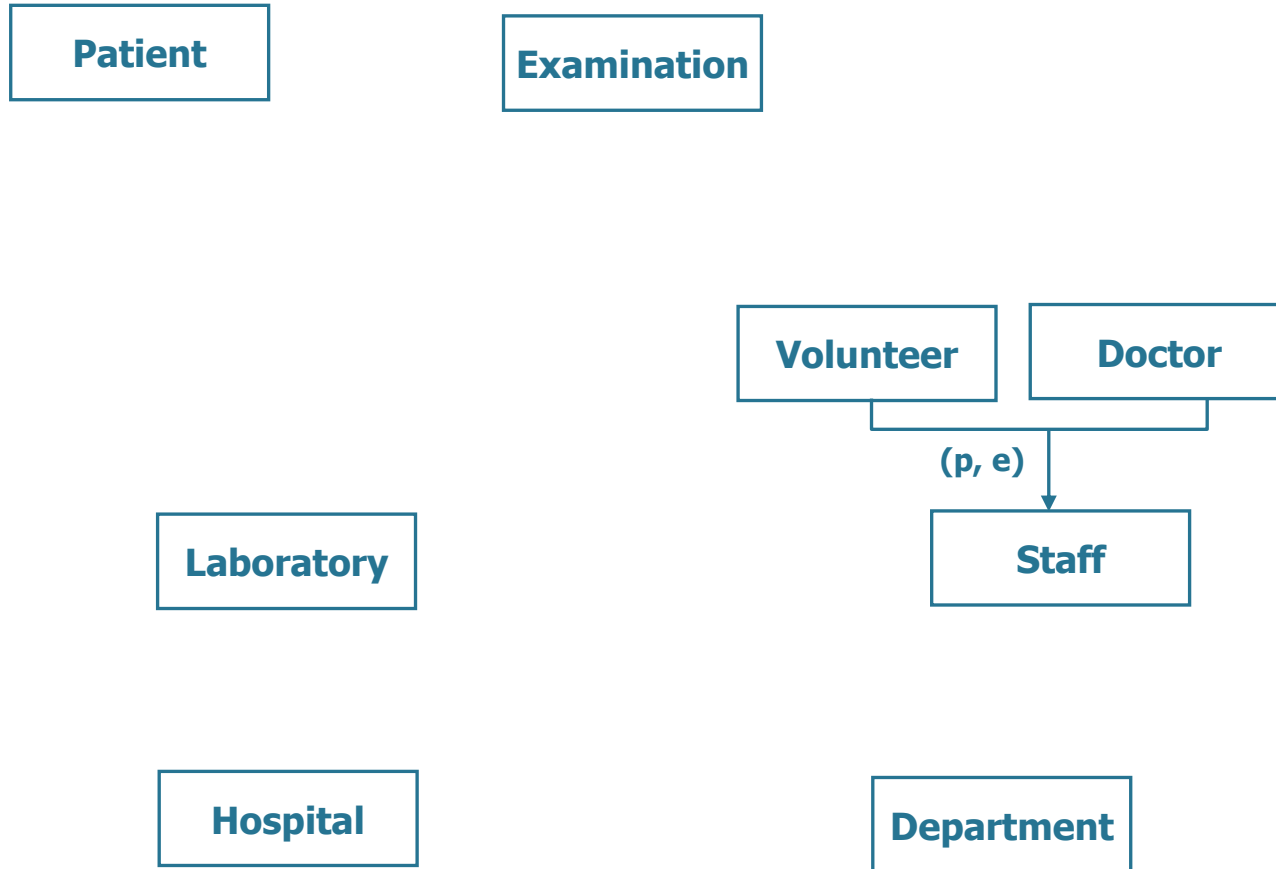
- Refinement of concepts
  - introduction of hierarchies
  - attributes definition
  - characterization of relationships with cardinality



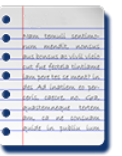
# Staff hierarchy

- The department staff is identified by a Social Security Number (SSN). Name, surname and home address are also known.
- Among the staff, for *doctors* the list of specializations achieved is known and for the *volunteers* the name of the association they belong to (if available) is stored.

# Staff hierarchy



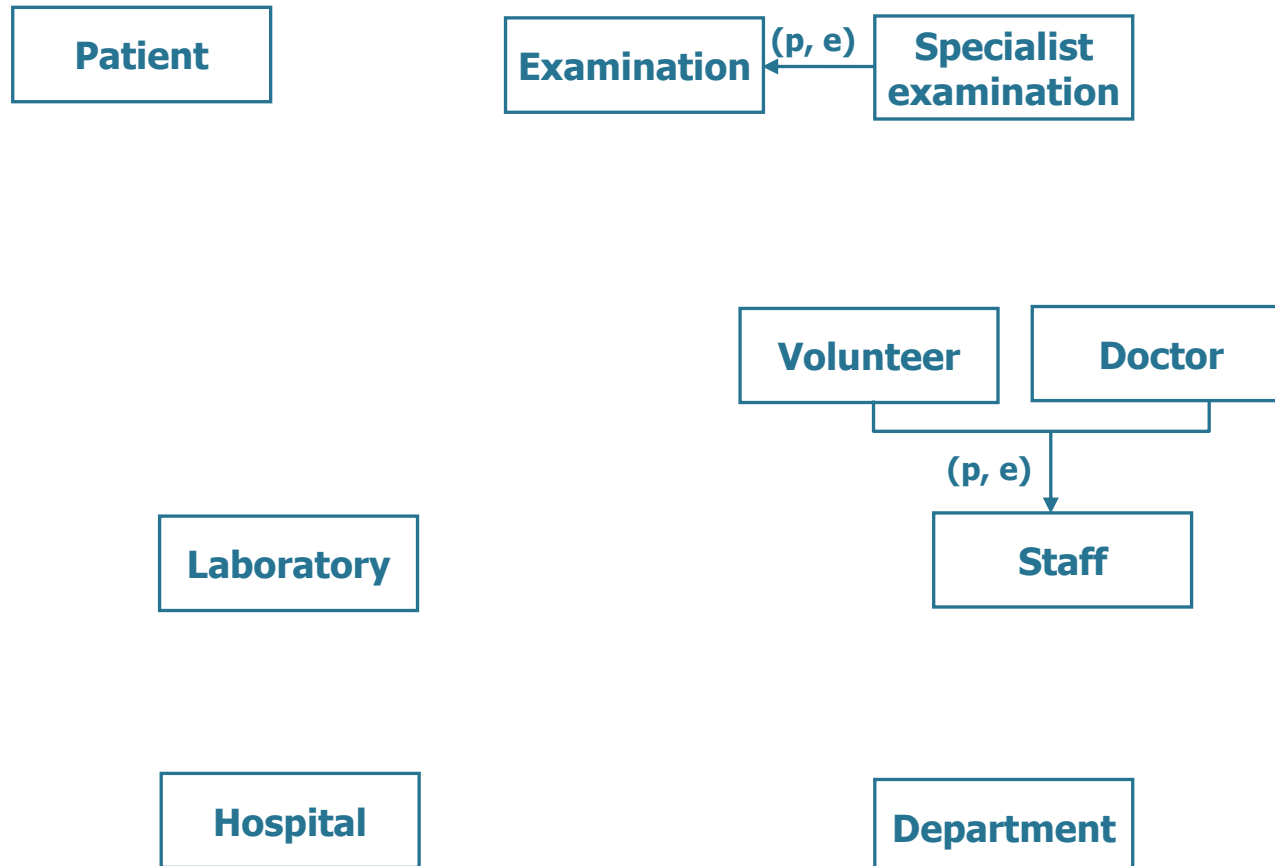
# Exam hierarchy



- The medical examinations that can be performed are characterized by a numerical code and a textual description (e.g. radiography, etc.). For *specialist examinations*, the doctor who carries out the visit and a description of the diet to follow (if necessary) are also stored.

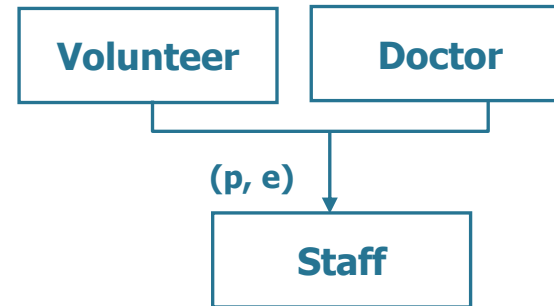
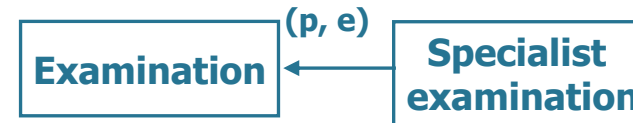
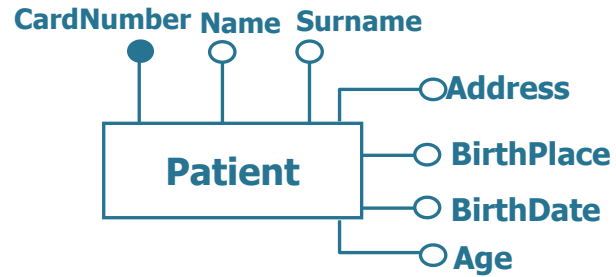


# Exam hierarchy



# Refinement of the Patient entity

*Each patient is characterized by health card number, name, surname, address, date of birth, place of birth and age.*



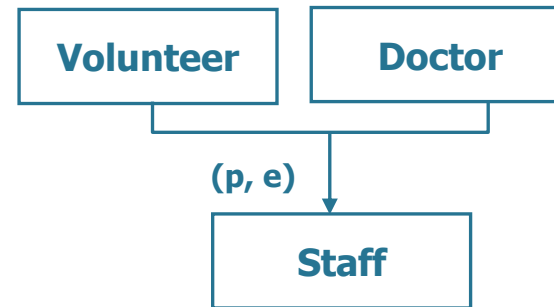
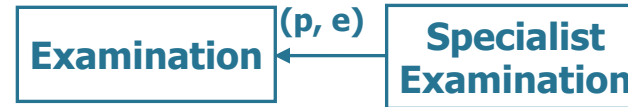
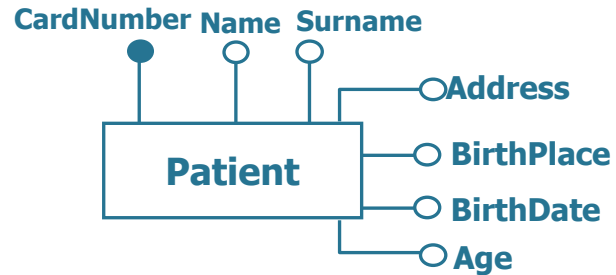
# Date of birth and Age Attributes

---

- The Age attribute is redundant because it can be easily calculated starting from the date of birth (BirthDate)
- This information must be attached to the conceptual model documentation
  - Age derivation rule from BirthDate:  $\text{Age} = \text{Year}(\text{Today}()) - \text{BirthDate}$
- Elimination of the Age attribute will be evaluated during the restructuring phase of the ER scheme

# Refinement of the Hospital entity

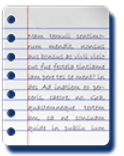
*ASL hospitals are characterized by a numerical code, name and address.*



# Database design

---

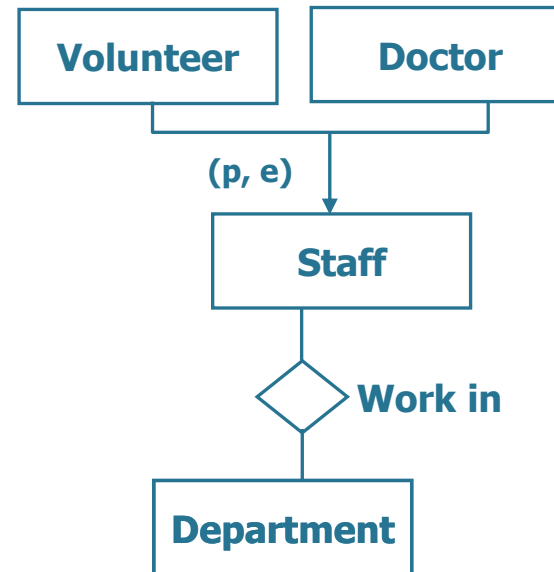
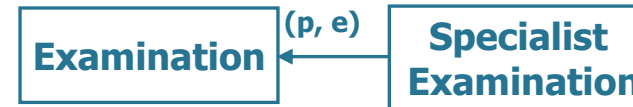
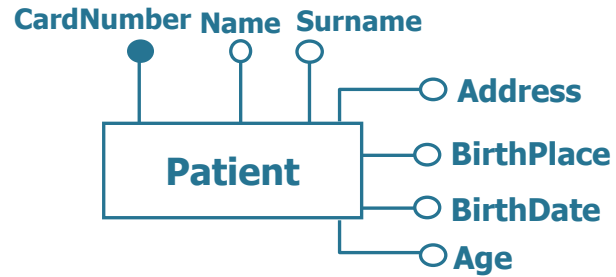
## Model refinement (II)



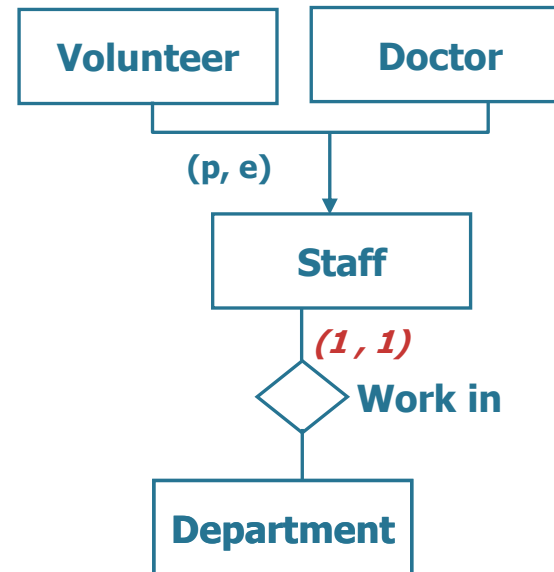
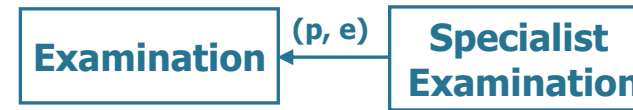
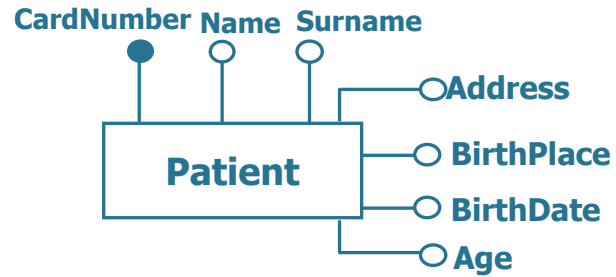
# Relationship between staff and department

- *The department staff is identified by a Social Security Number (SSN). Name, surname and address are also known. Among the staff, for doctors the list of specializations achieved is known and for volunteers the name of the association they belong to (if available) is stored.*

# Relationship between staff and department

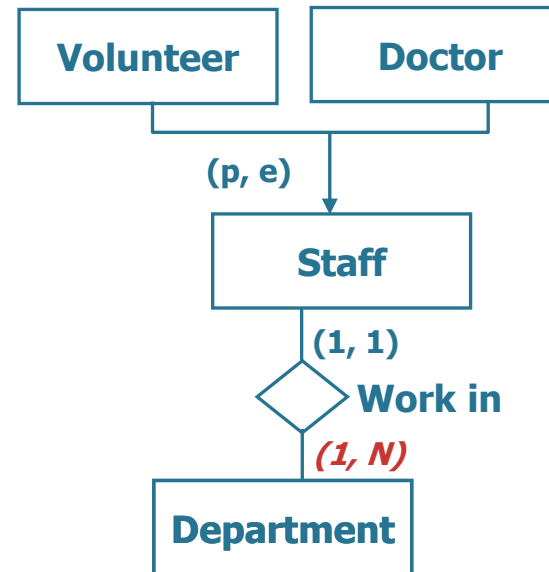
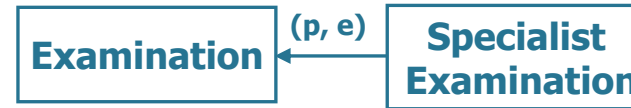
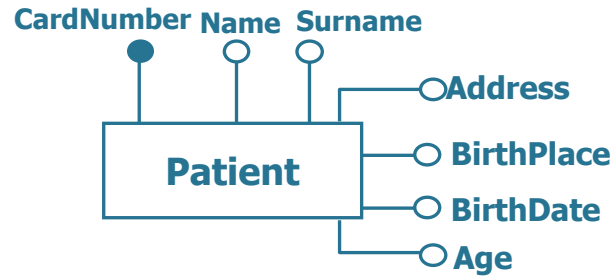


# Cardinality of the *work in* Relationship

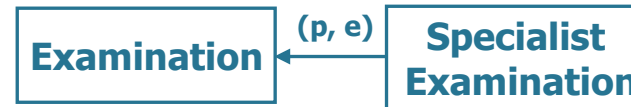
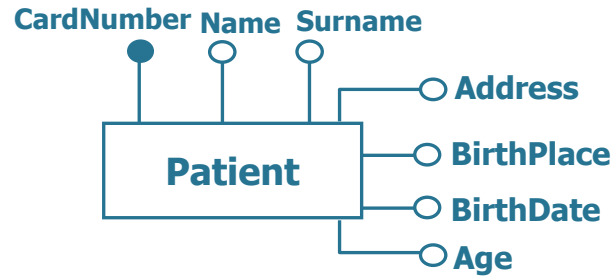




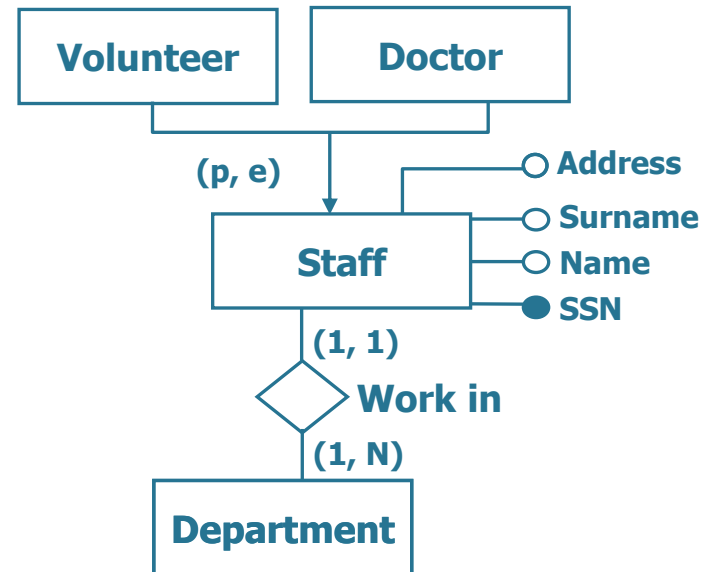
# Cardinality of the *work in* Relationship



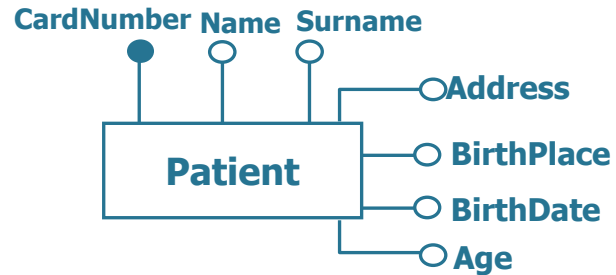
# Refinement of the Staff entity



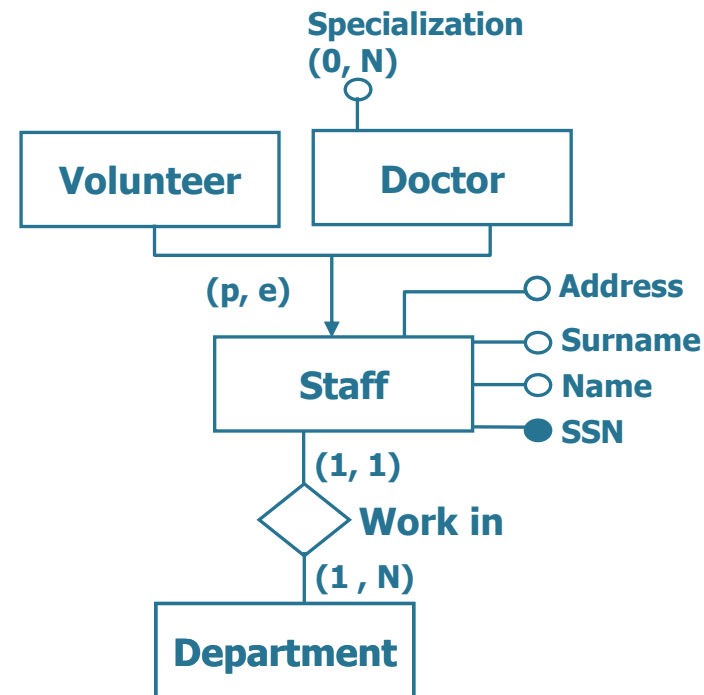
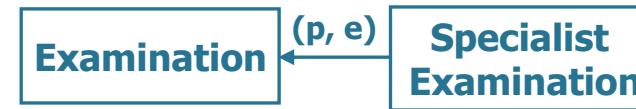
*The department staff is identified by a Social Security Number (SSN). Name, surname and address are also known.*



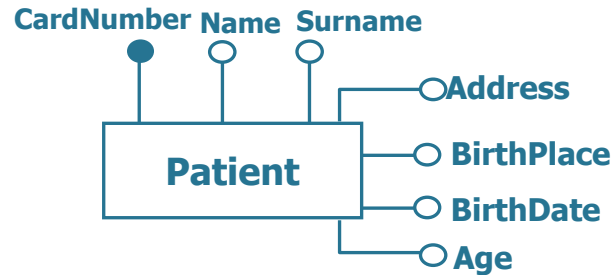
# Refinement of the Doctor and Volunteer entities



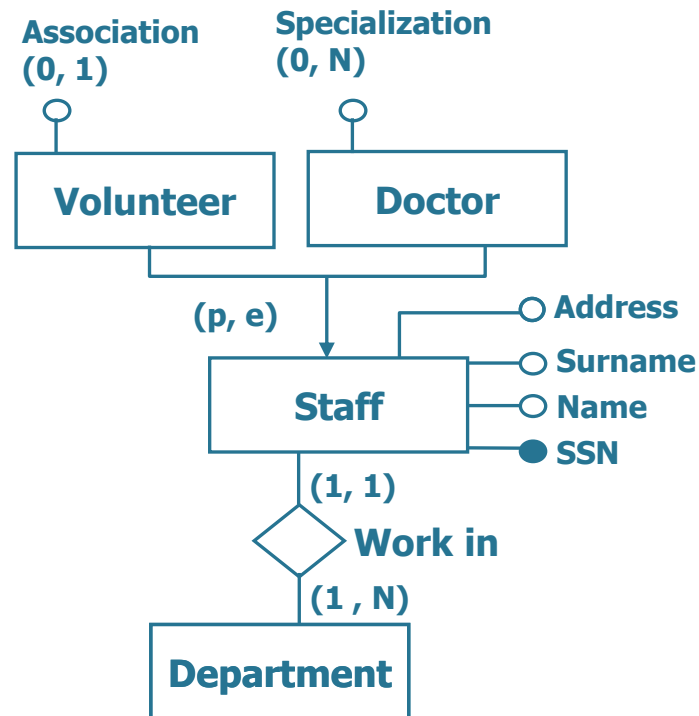
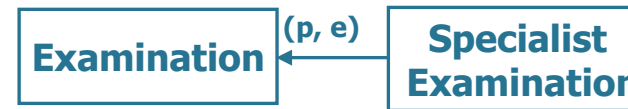
*Among the staff, for doctors the list of specializations achieved is known and for volunteers the name of the association they belong to (if available) is stored.*



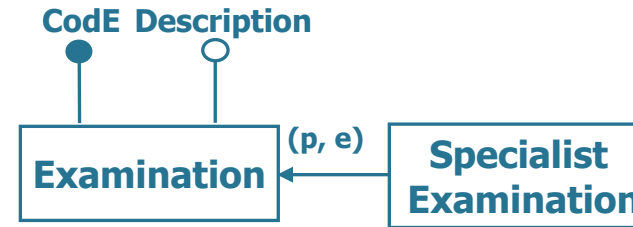
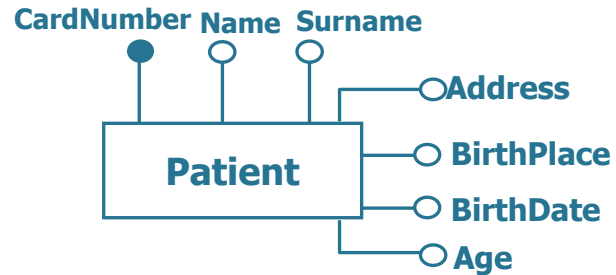
# Refinement of the Doctor and Volunteer entities



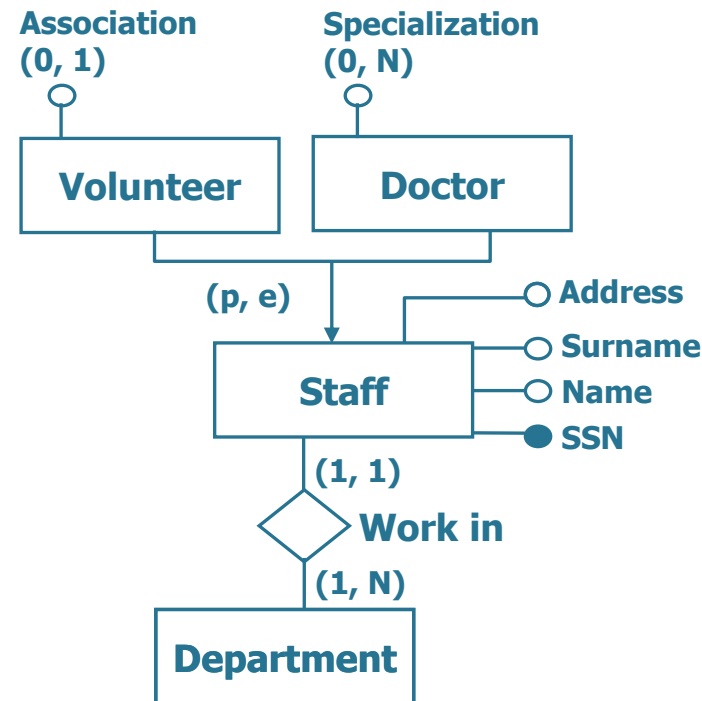
*Among the staff, for doctors the list of specializations achieved is known and for volunteers the name of the association they belong to (if available) is stored.*



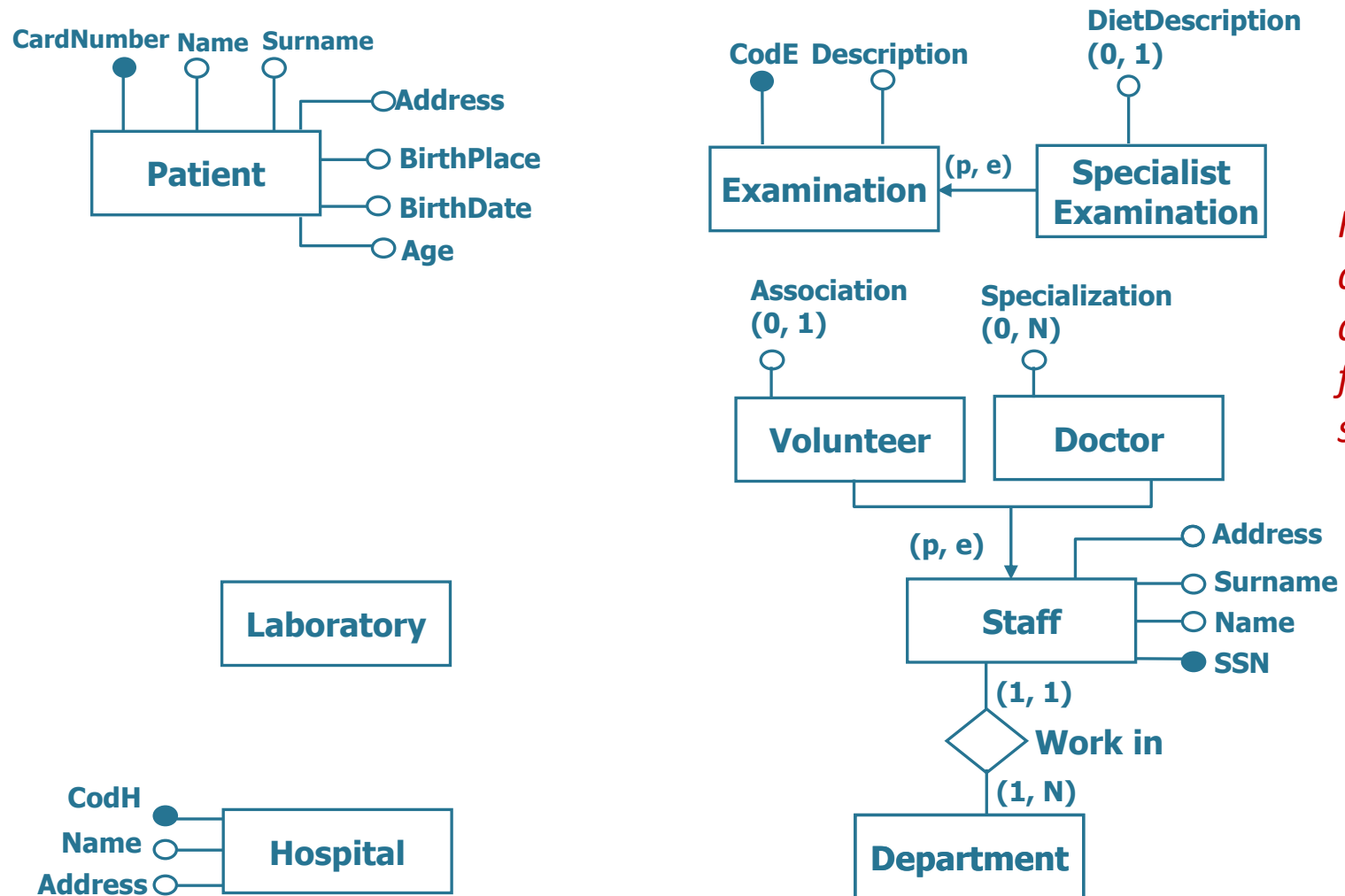
# Refinement of the Examination entity



*The medical examinations that can be performed are characterized by a numerical code and a textual description (e.g. X-ray exam, etc.).*

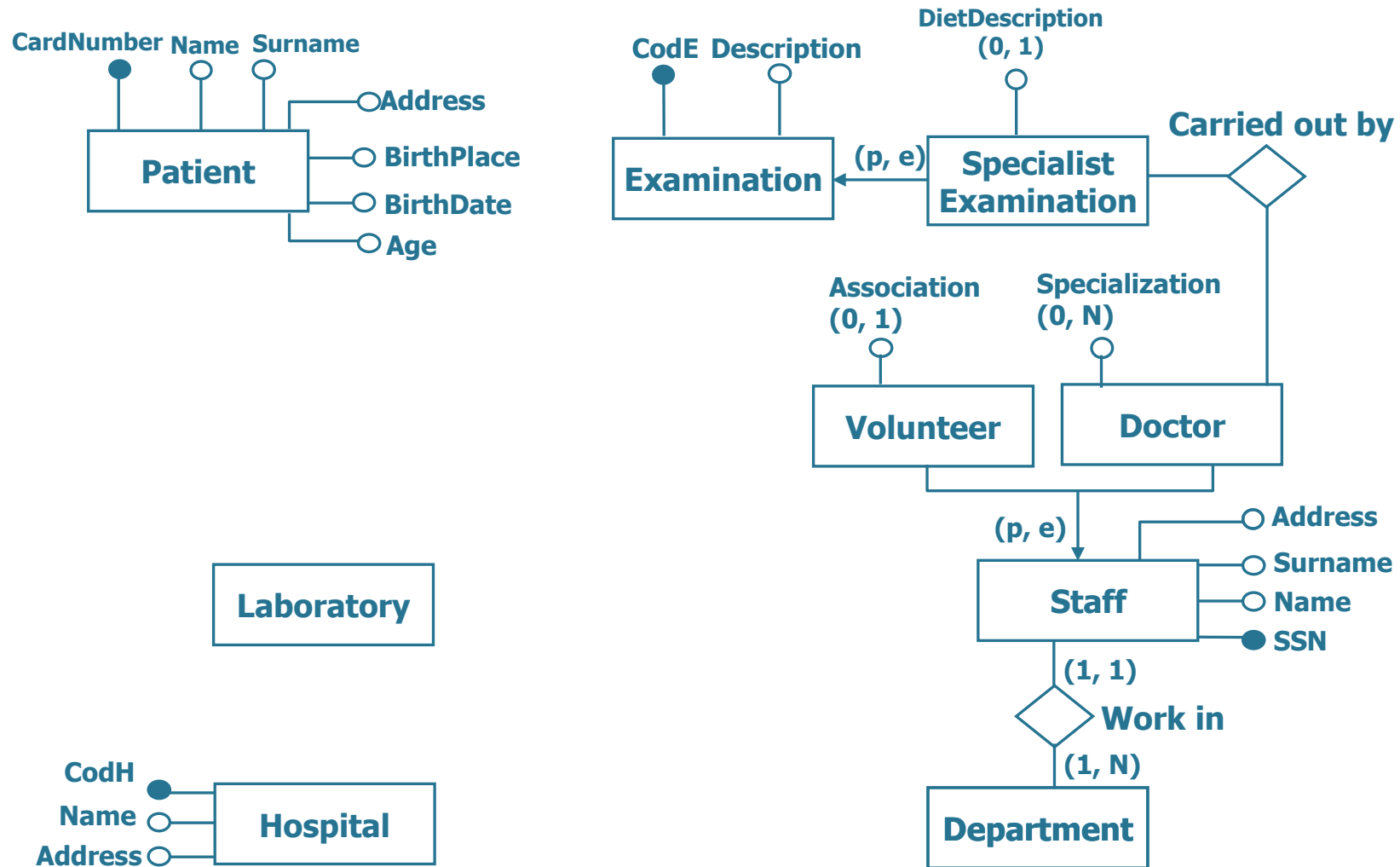


# Refinement of the Specialistic Examination entity

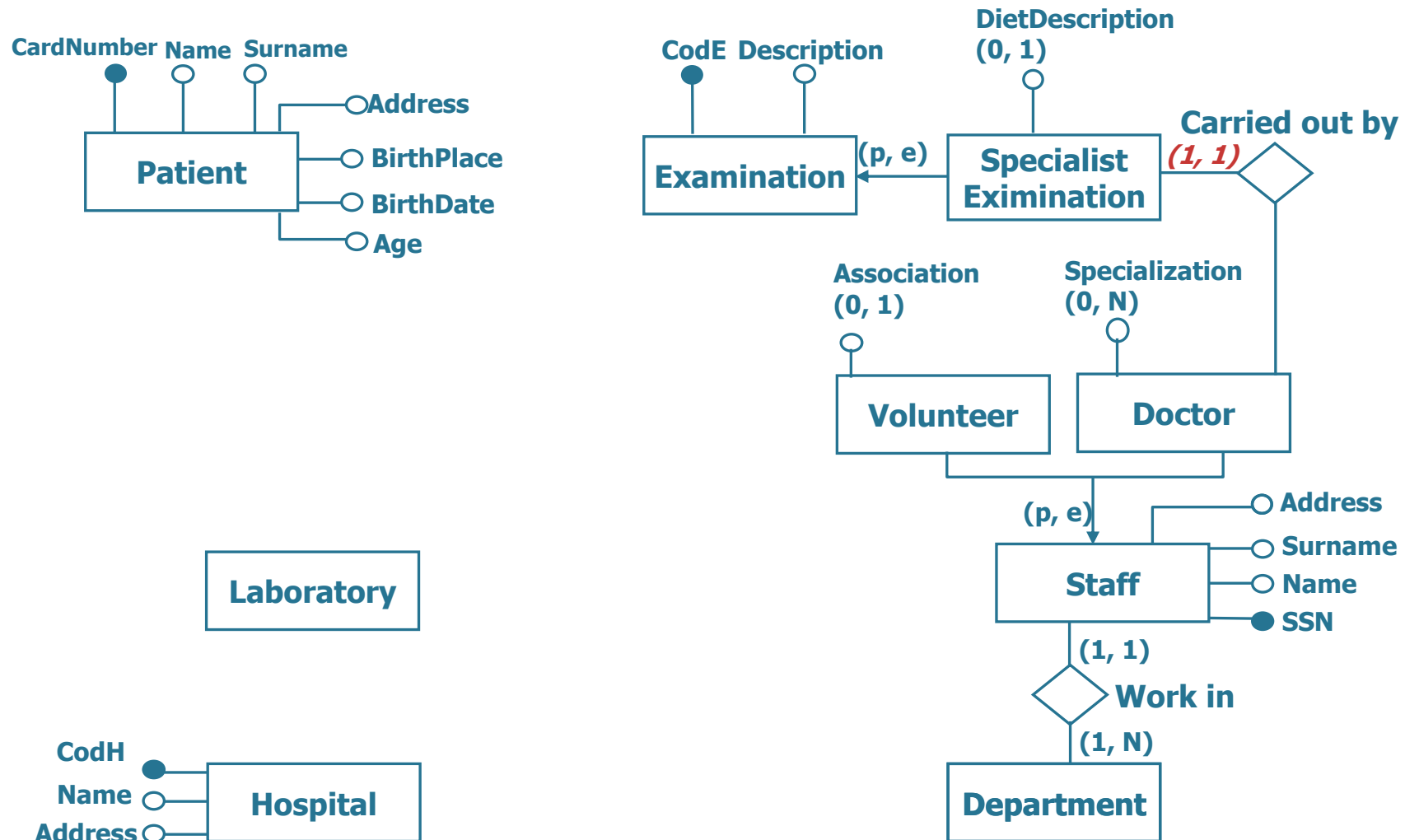


*For specialist examinations, the doctor who carries out the visit and a description of the diet to follow (if necessary) are also stored.*

# Refinement of the Specialistic Examination entity

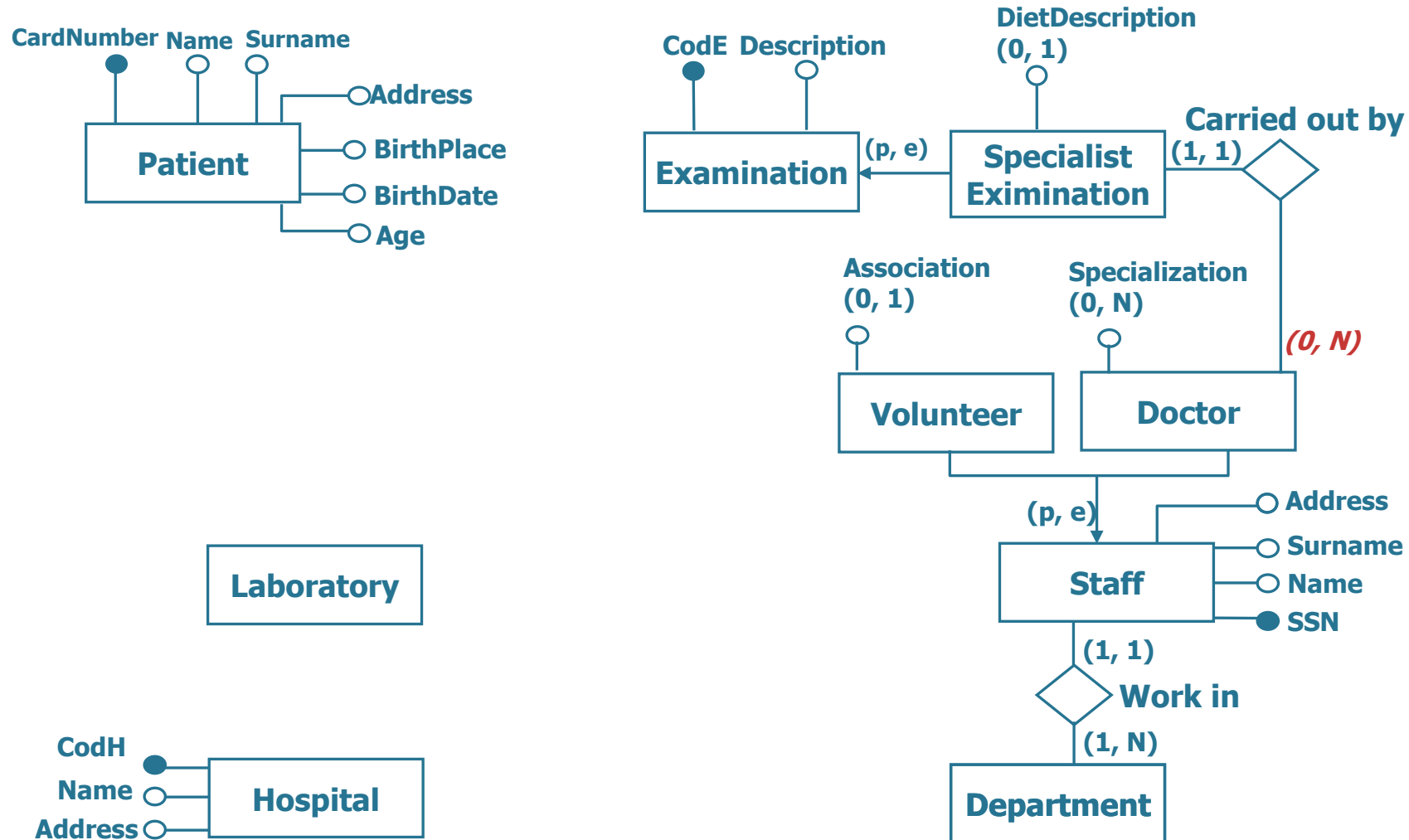


# Cardinality of the *Carried out by* relationship





# Cardinality of the *Carried out by* relationship

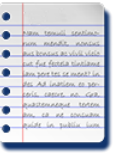


# Database design

---

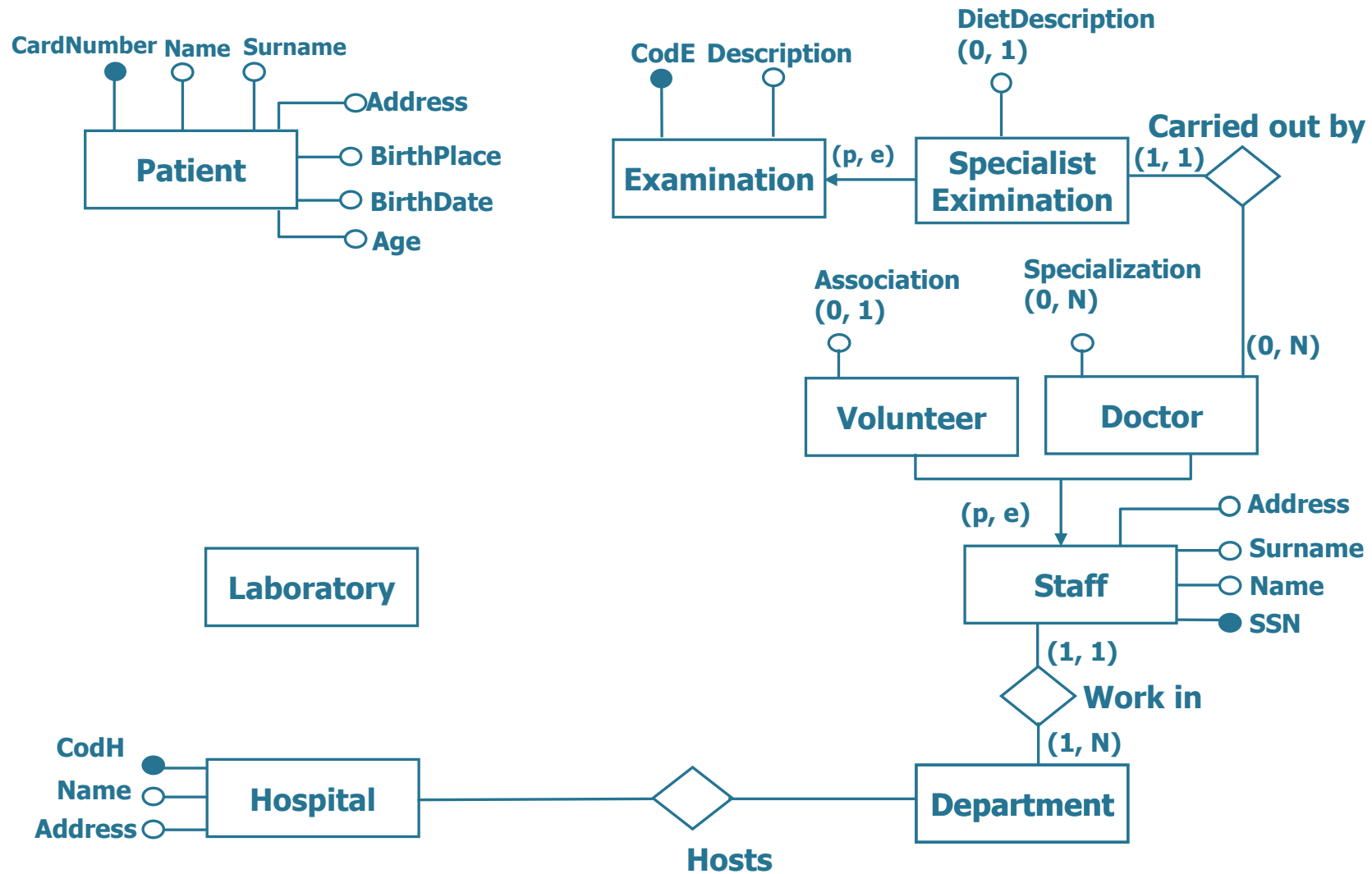
Model refinement (III)

# Relationship between Department and Hospital

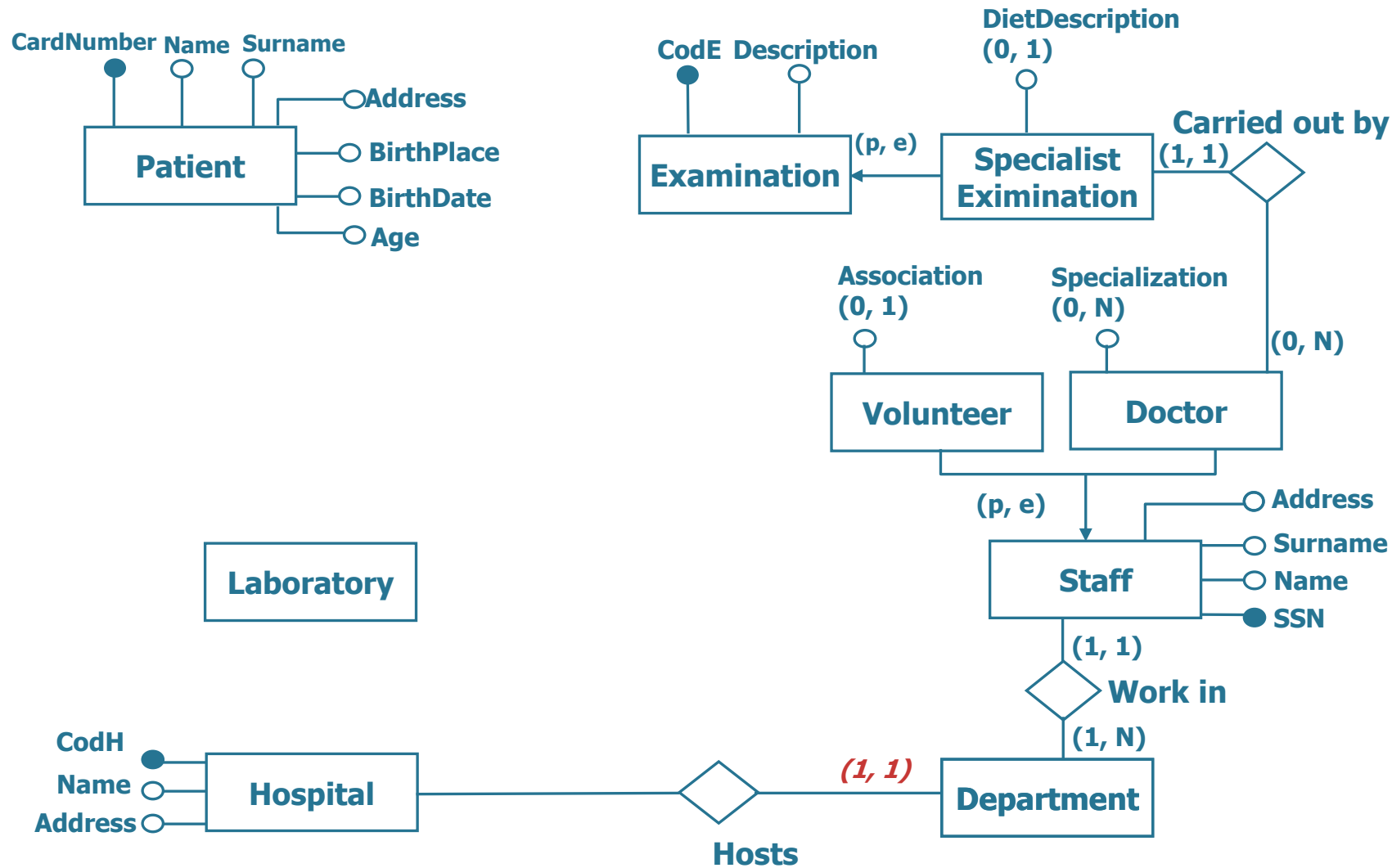


- Each hospital is divided into departments identified by a unique numeric code within the hospital and characterized by the department name and the telephone number.

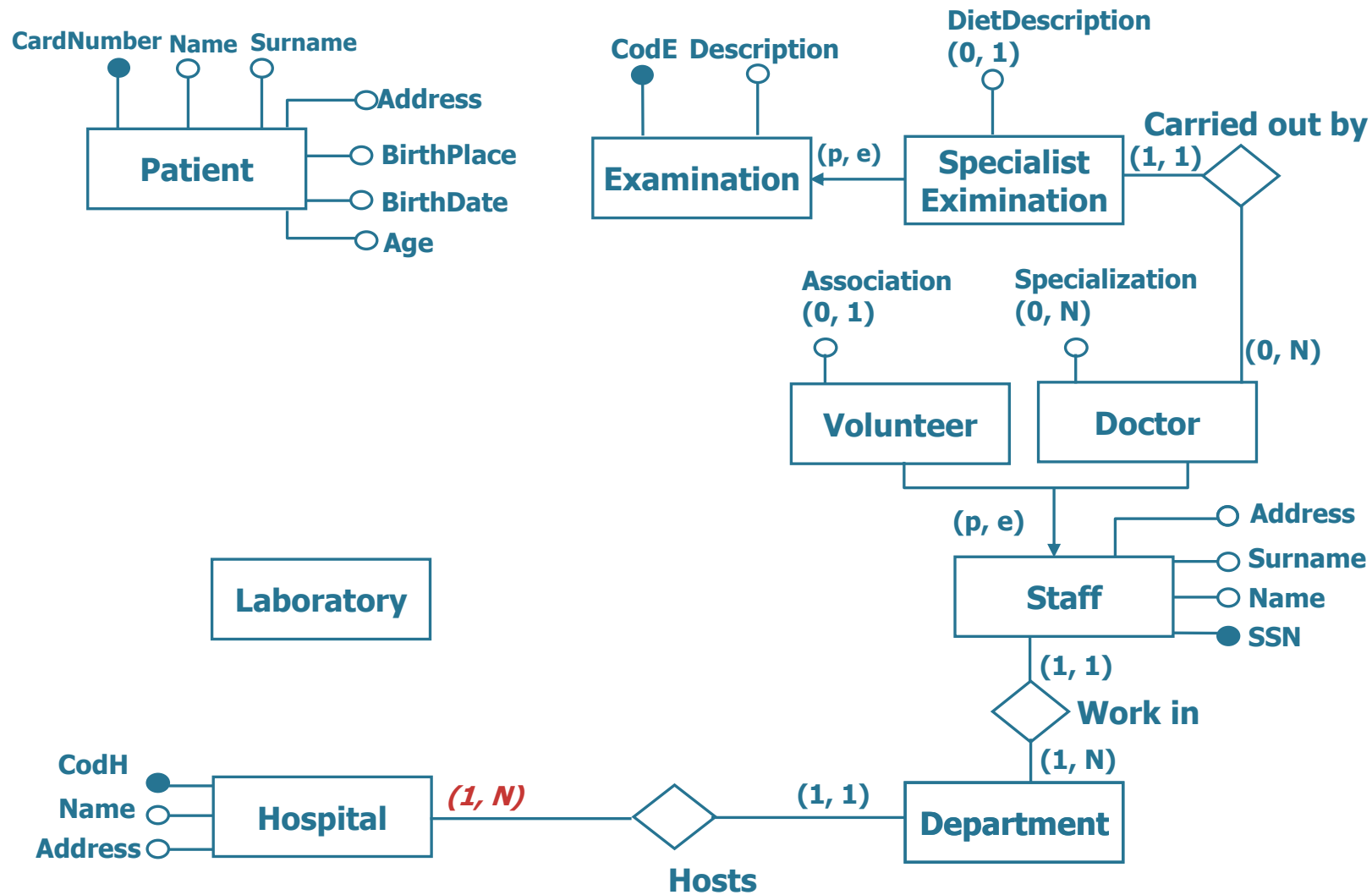
# Relationship between Department and Hospital

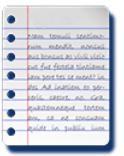


# Cardinality of the *Hosts* relationship



# Cardinality of the *Hosts* relationship

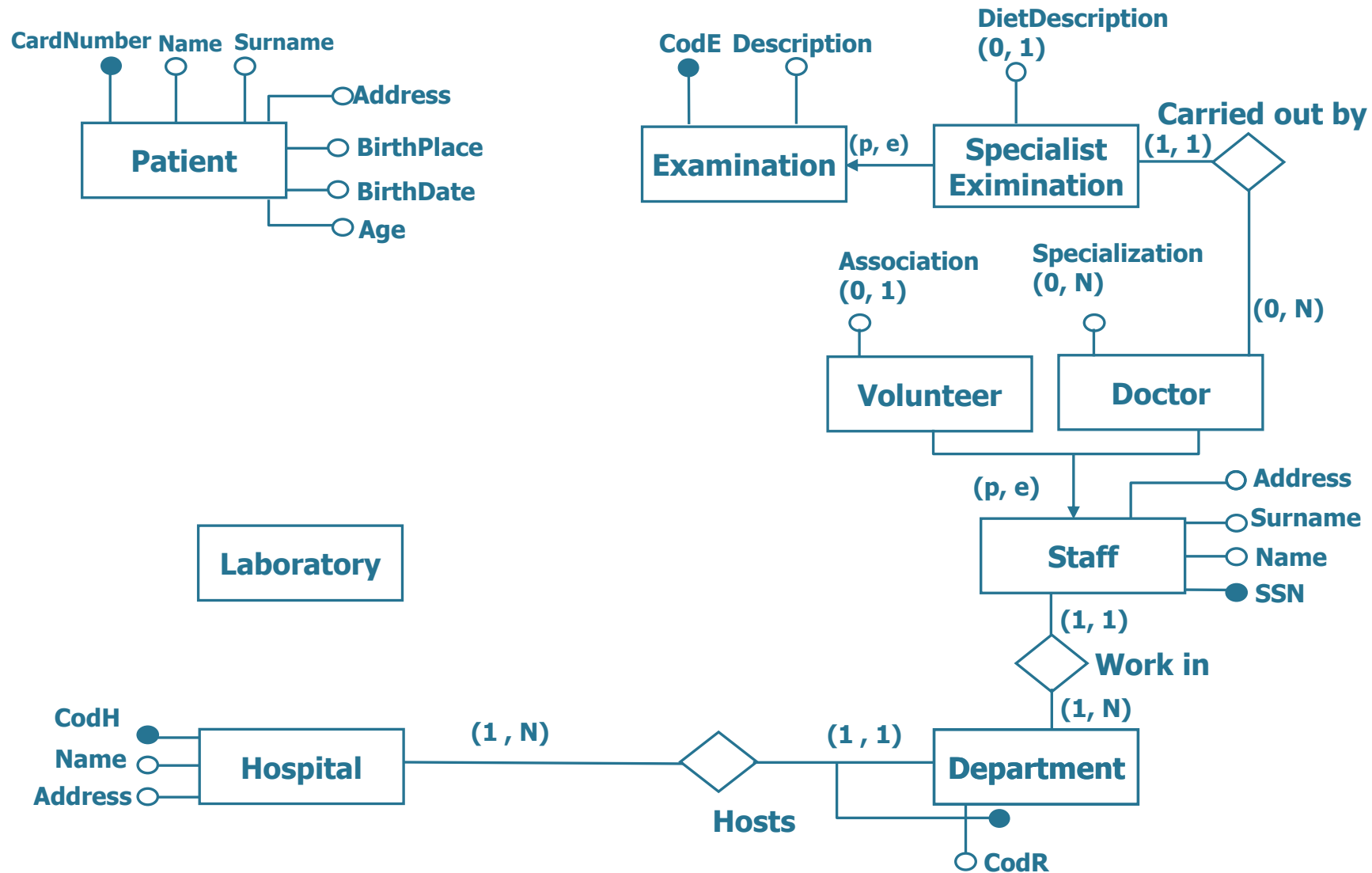




# Department entity identifier

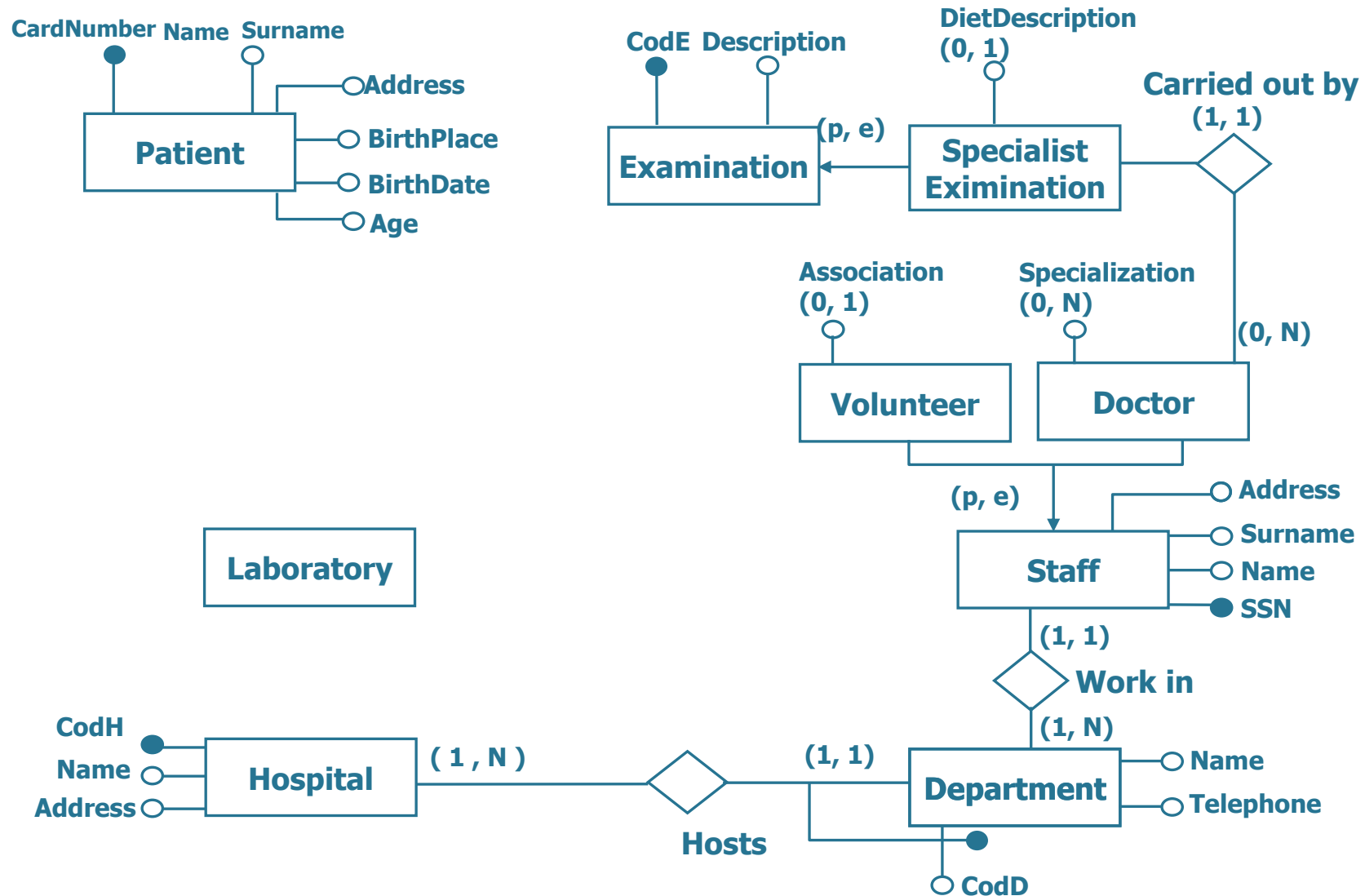
- Each hospital is divided into departments identified by a unique numeric code within the hospital and characterized by the department name and the telephone number.

# Department entity identifier

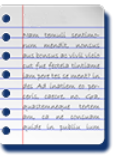




# Refinement of the Department entity

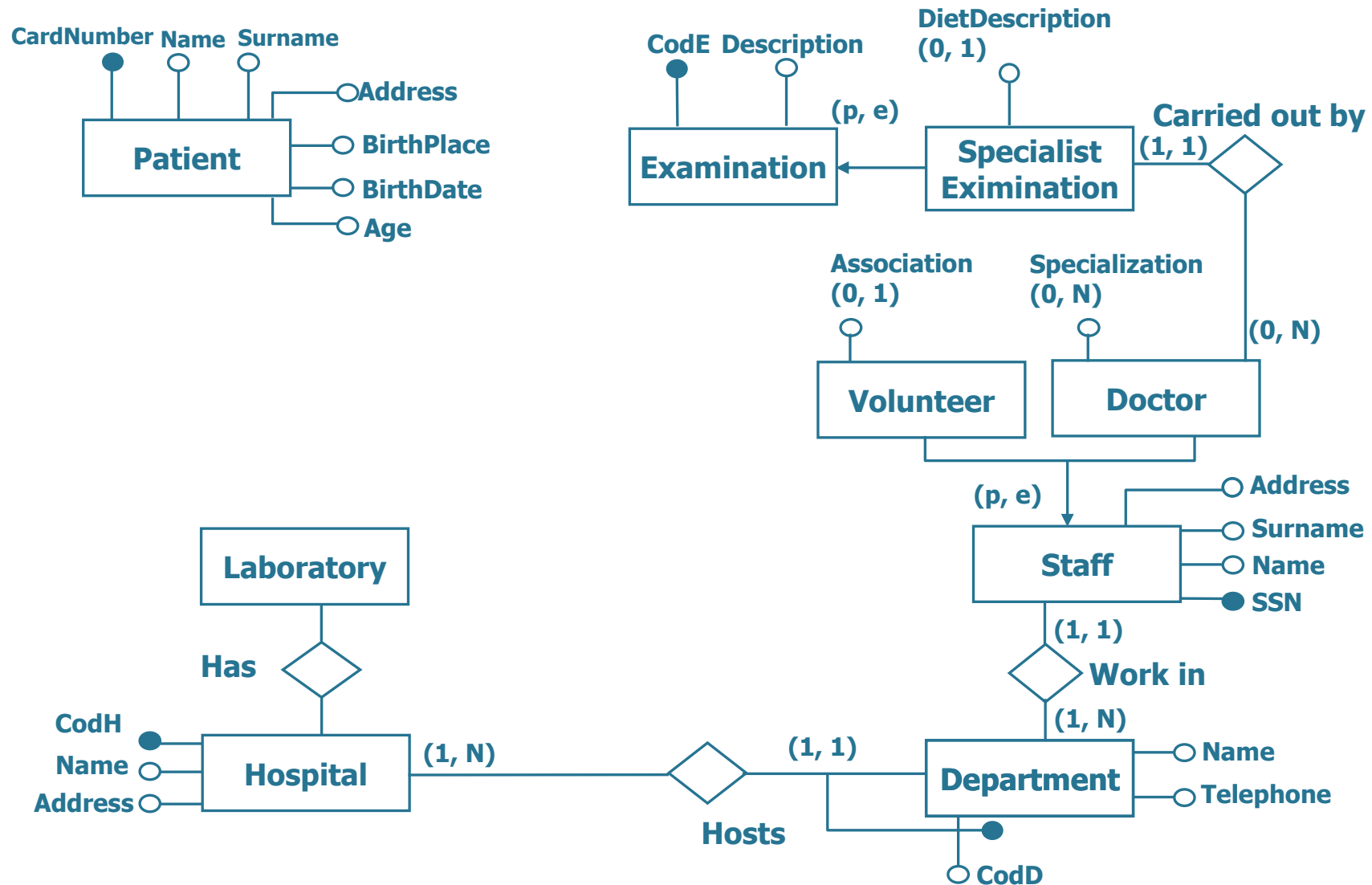


# Relationship between Laboratory and Hospital

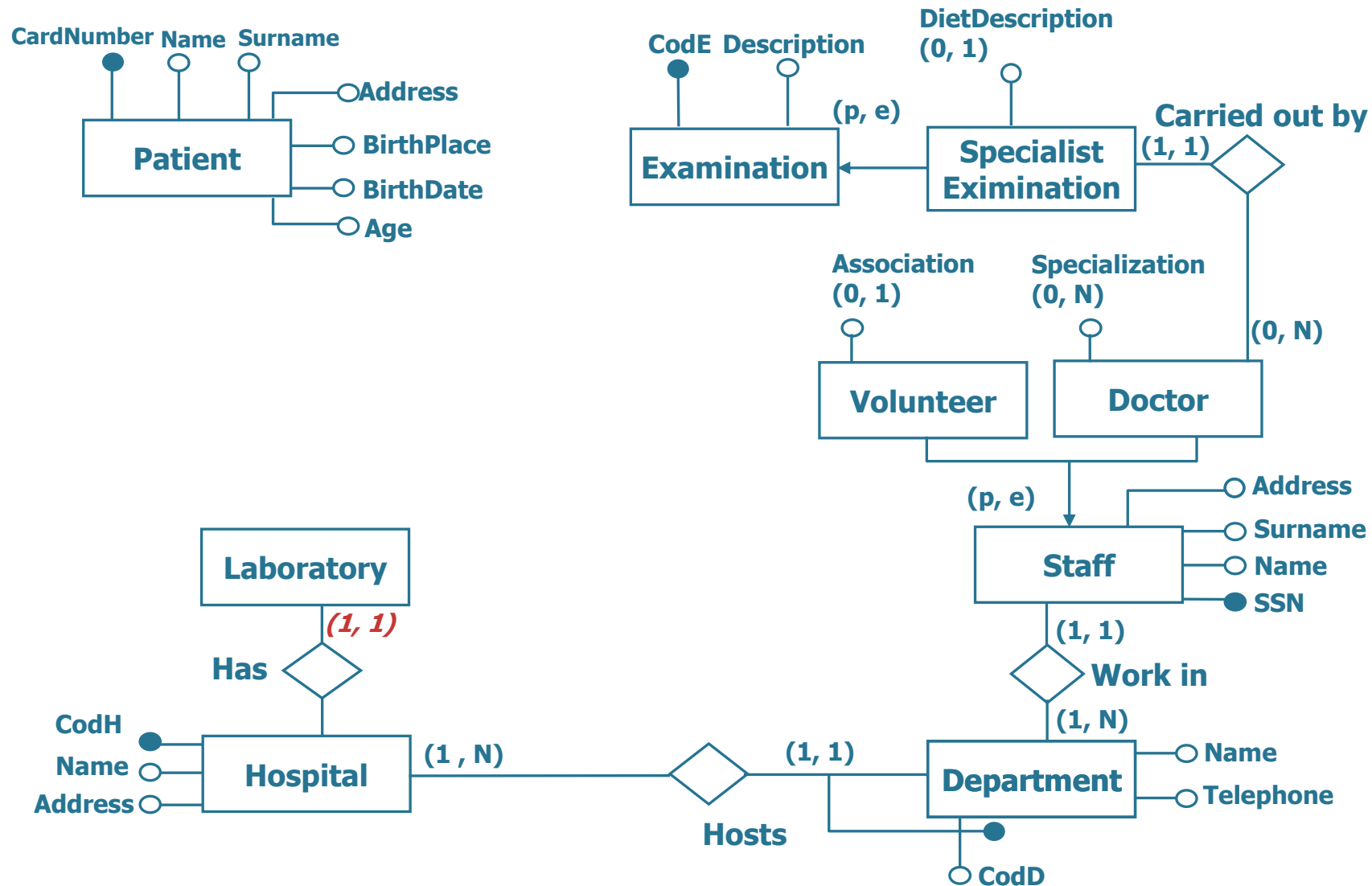


- The laboratories that perform the tests are identified by a unique code within the hospital and are characterized by the name of the laboratory, the location plan and the room number.

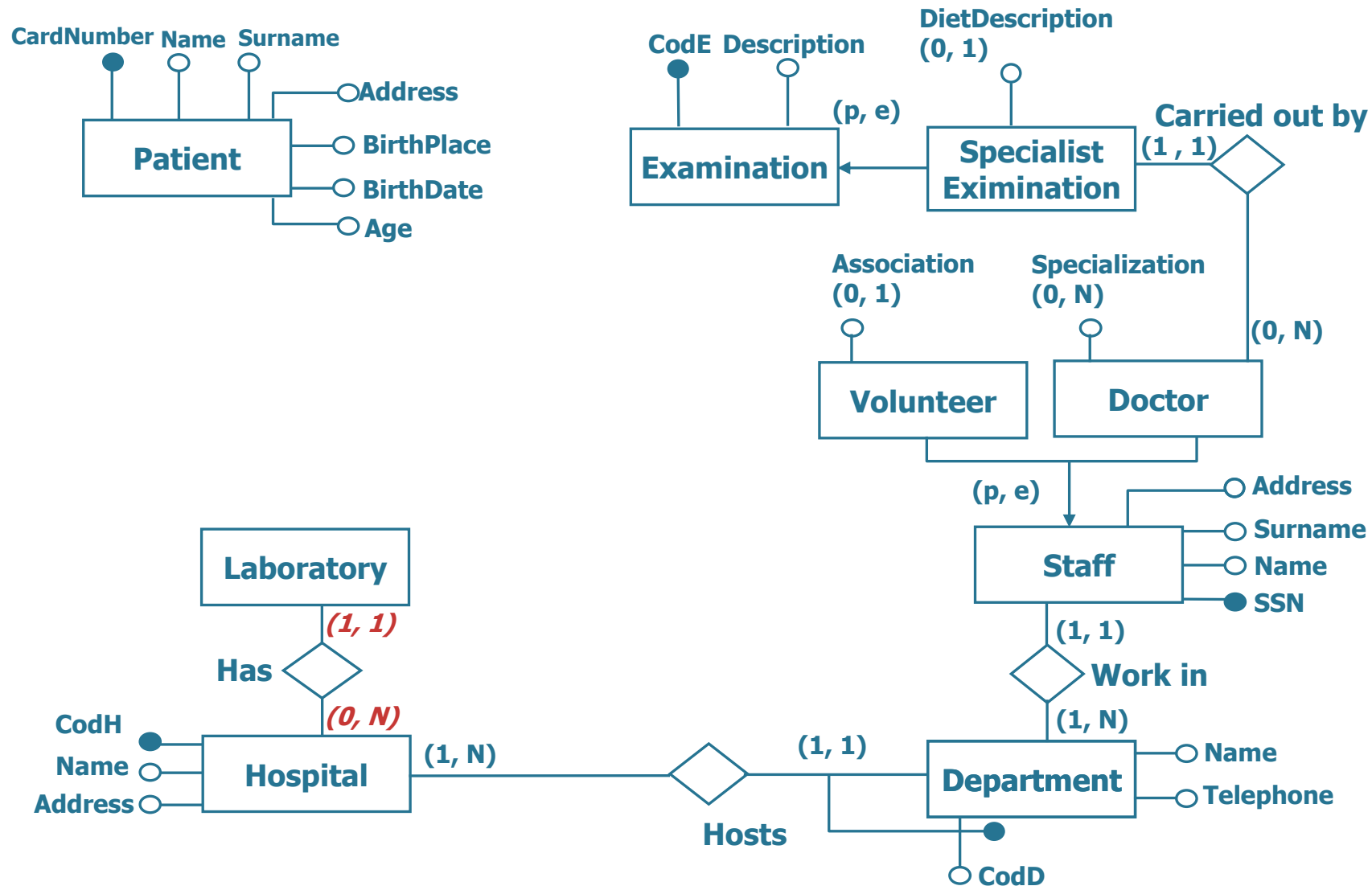
# Relationship between Laboratory and Hospital



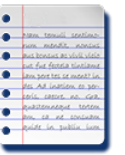
# Cardinality of the *Has* relationship



# Cardinality of the *Has* relationship

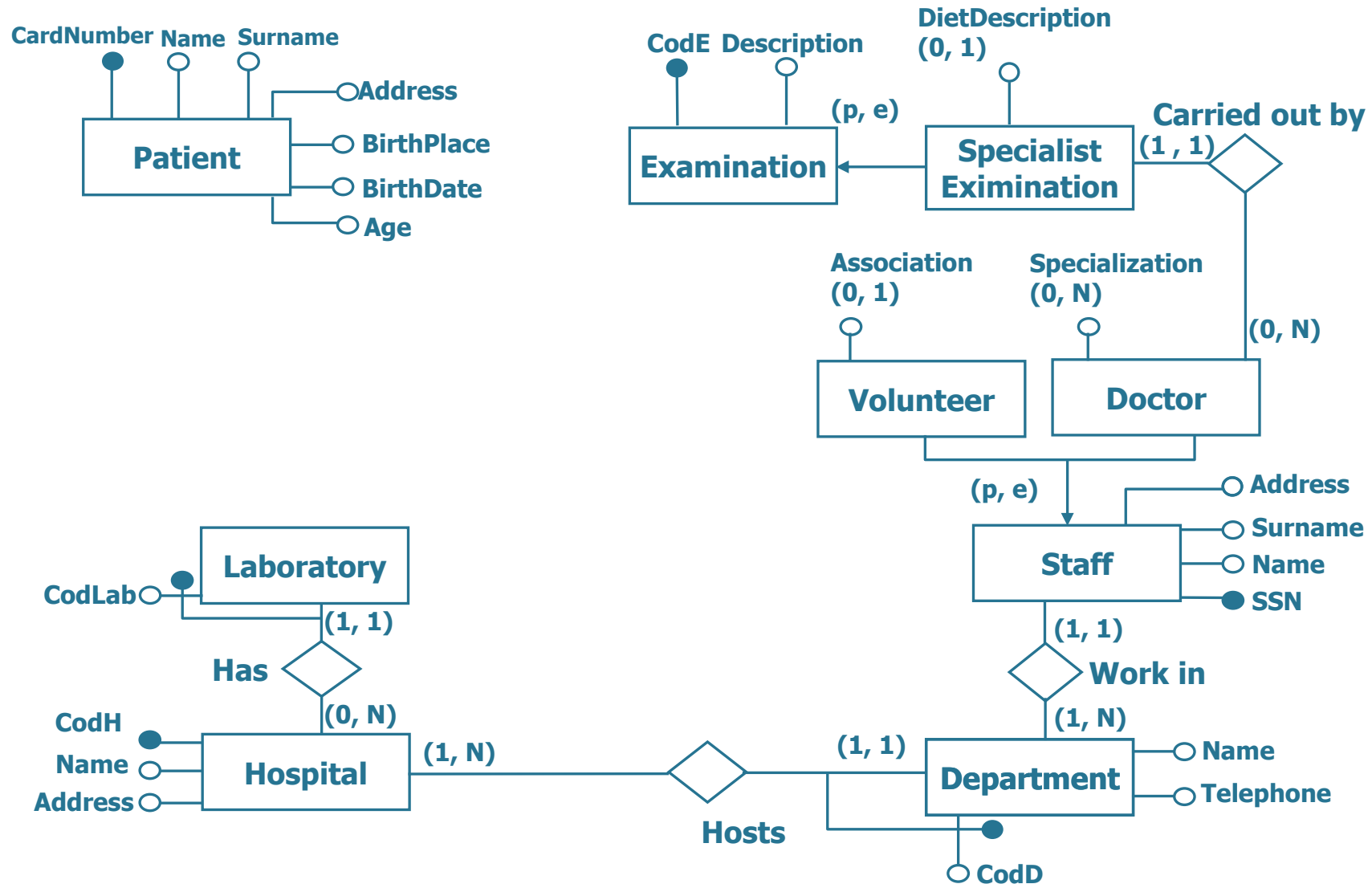


# Relationship between Laboratory and Hospital

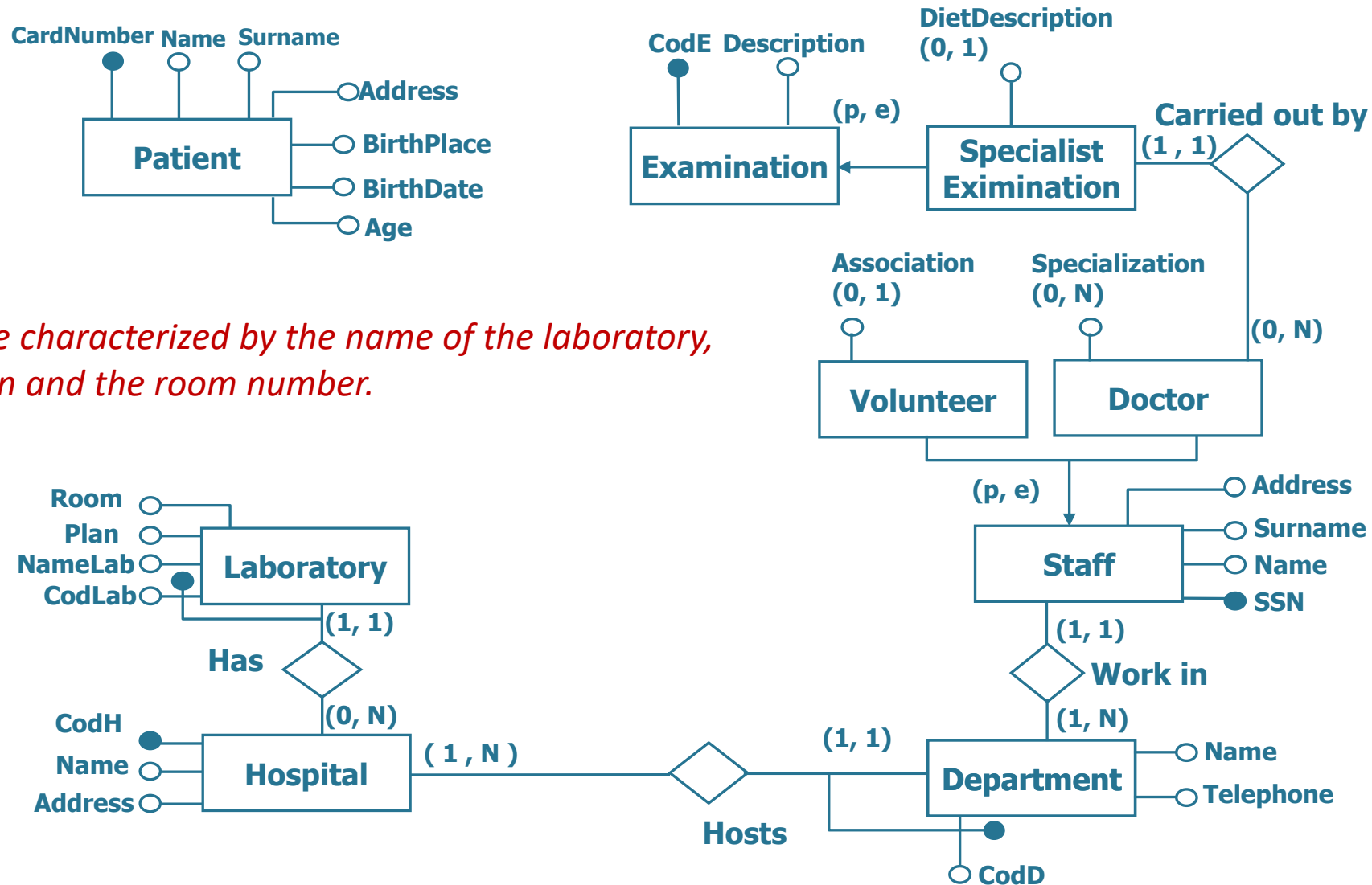


- The laboratories that perform the tests are identified by a unique code within the hospital and they are characterized by the name of the laboratory, the location plan and the room number.

# Laboratory entity identifier



# Refinement of the Laboratory entity



*Laboratories are characterized by the name of the laboratory, the location plan and the room number.*



# Database design

---

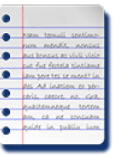
Representation of time (I)

# Time representation

---

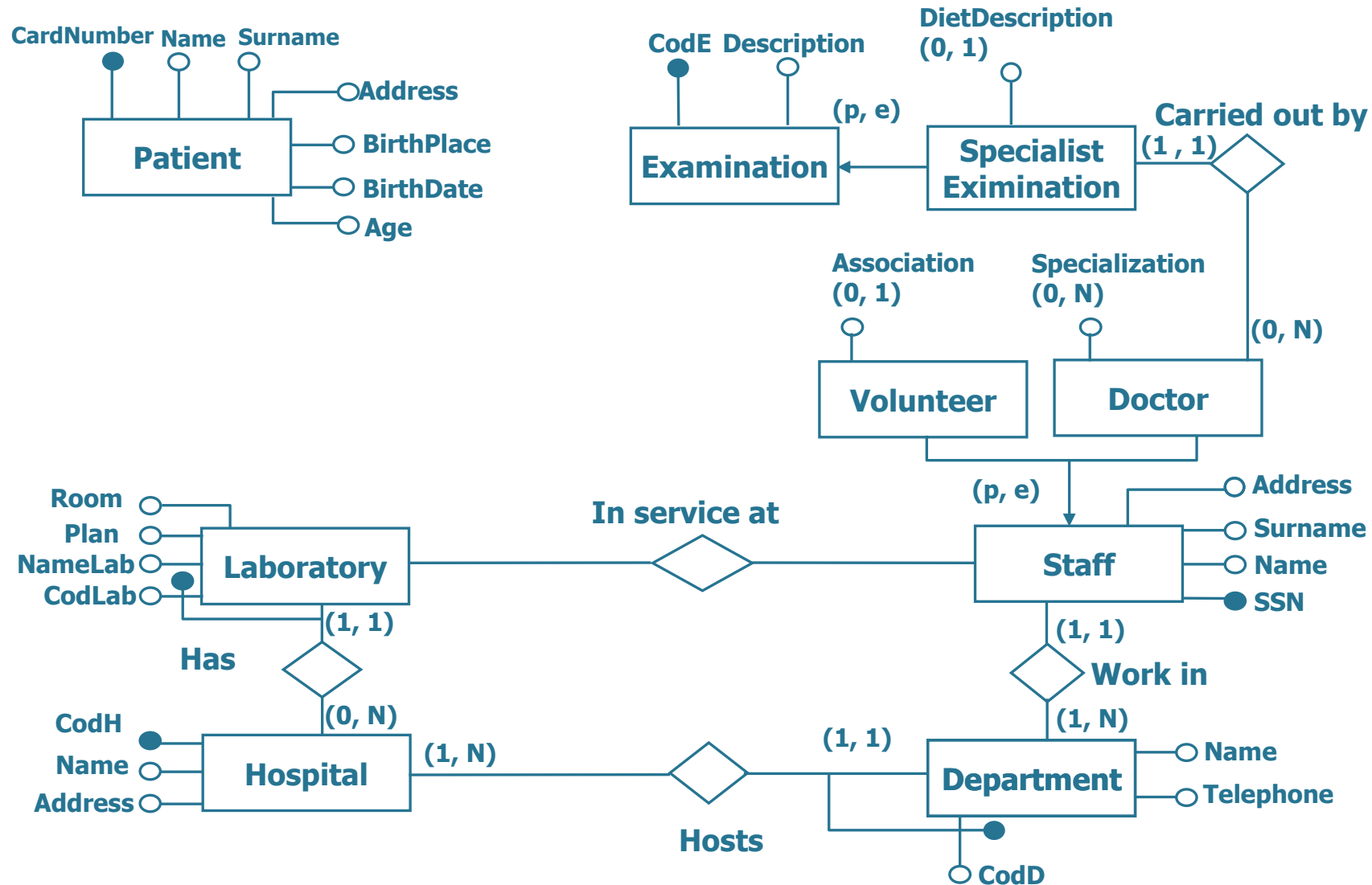
- It is necessary to explicitly represent time-related information in the case of
  - representation of events
  - changes in the information content of entities or time attributes
- Various patterns are possible
  - by means of N-ary relationship with a time entity
  - through historicized entities
  - through binary relationships with a time entity

# Relationship between Staff and Laboratory

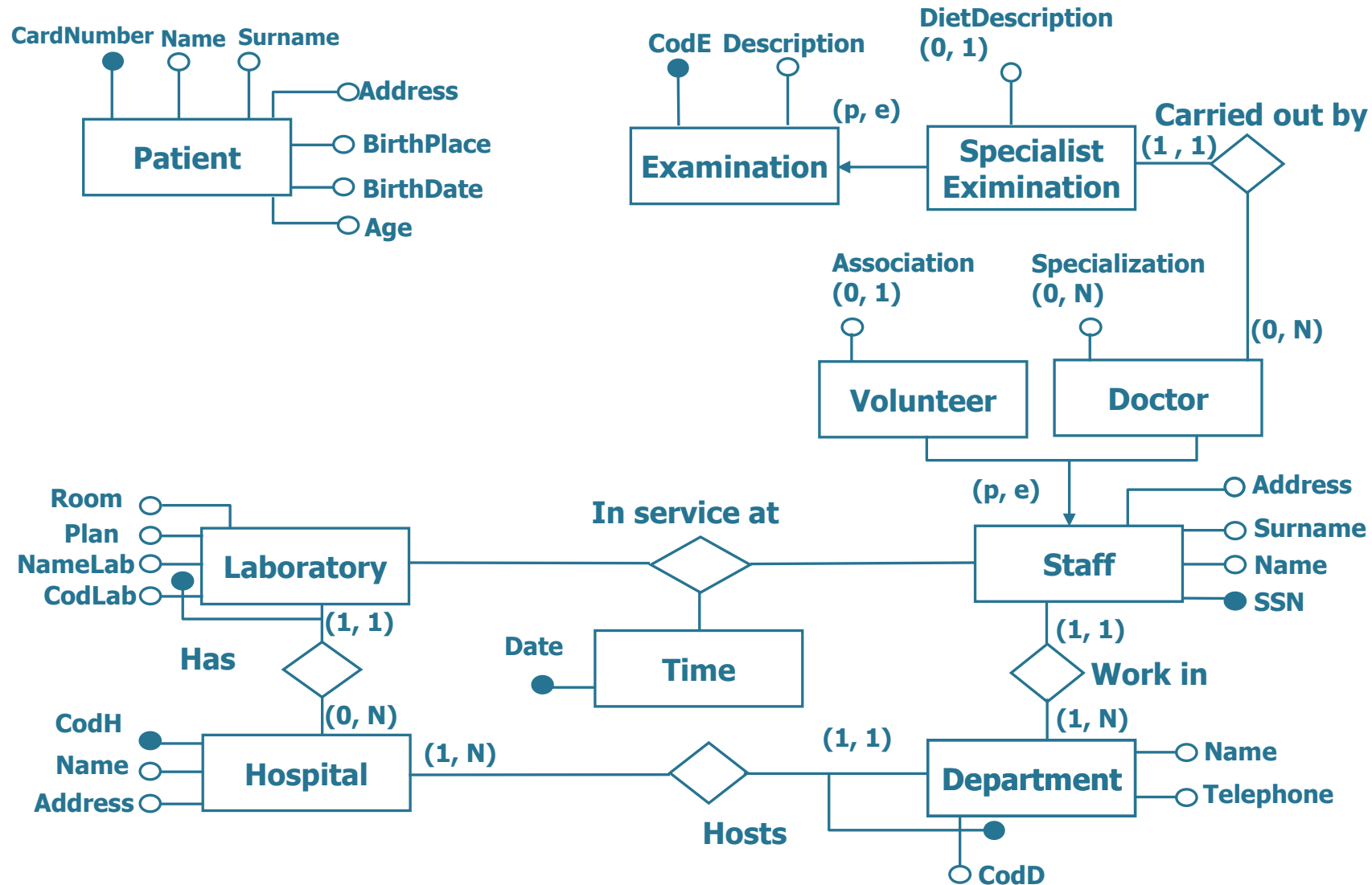


- For each member of the laboratory staff, the days and laboratories in which he/she works are stored. It should be borne in mind that during the same day each member of the staff can work in several laboratories.

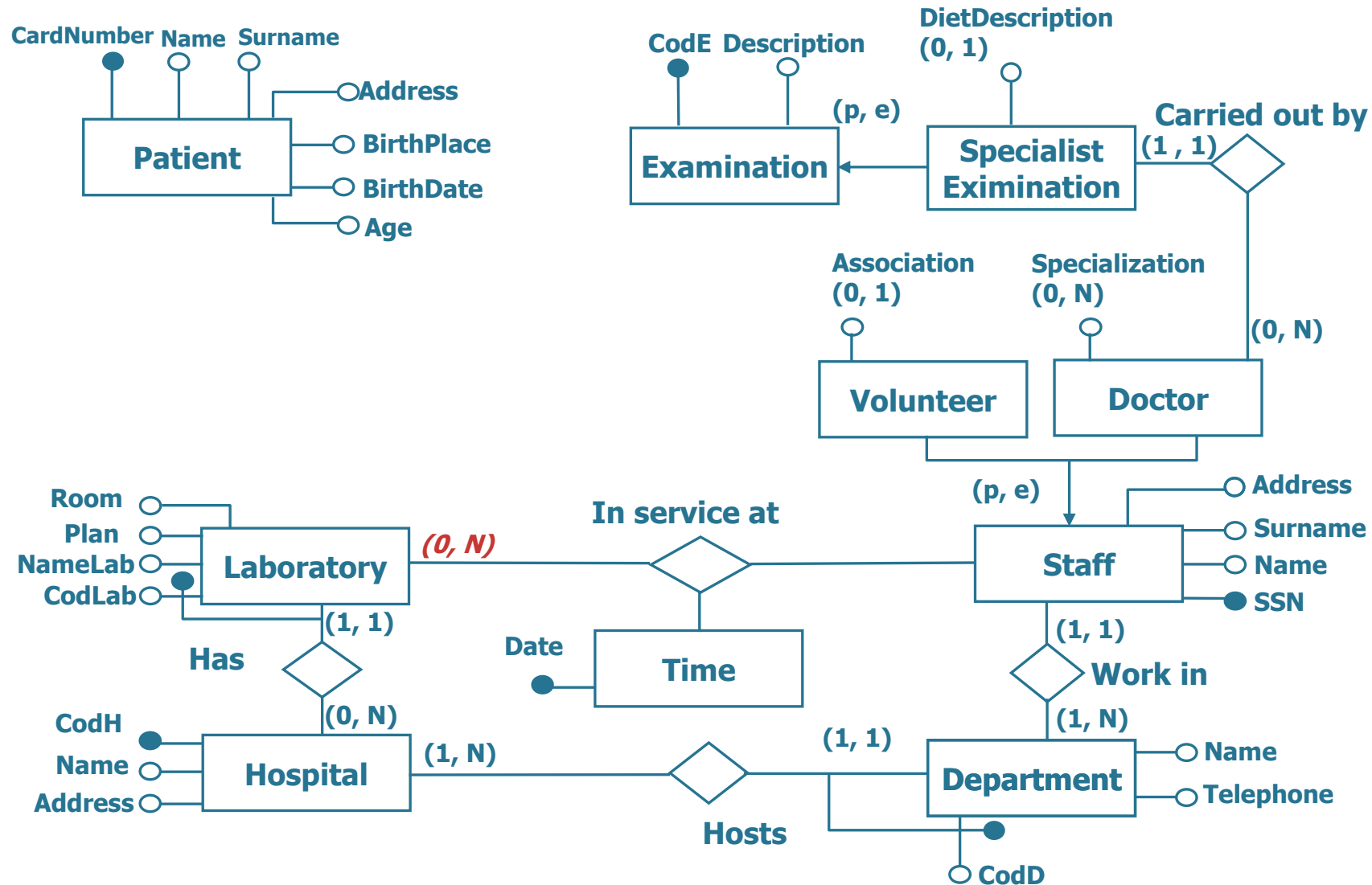
# Relationship between Staff and Laboratory



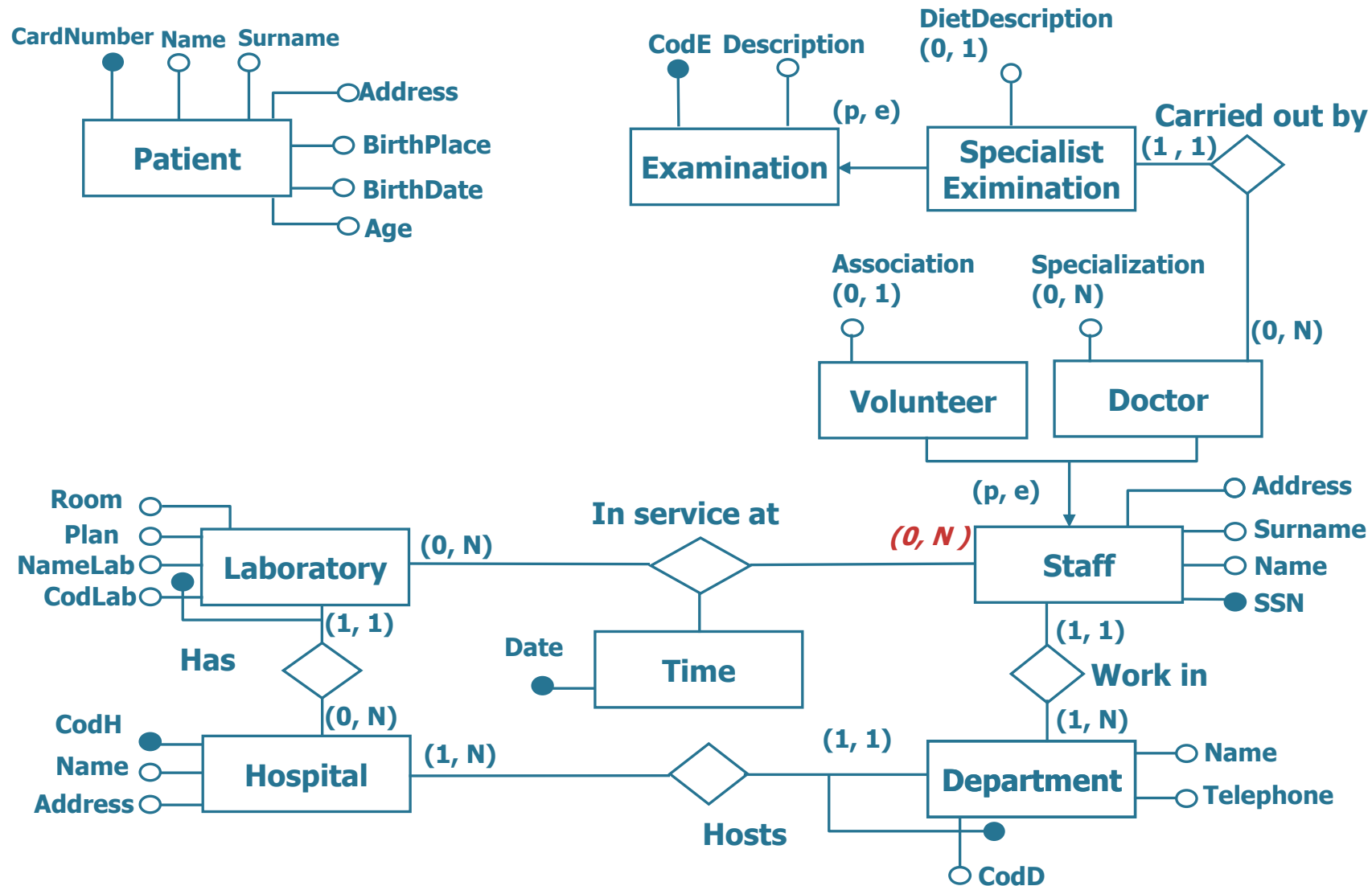
# Relationship between Staff and Laboratory



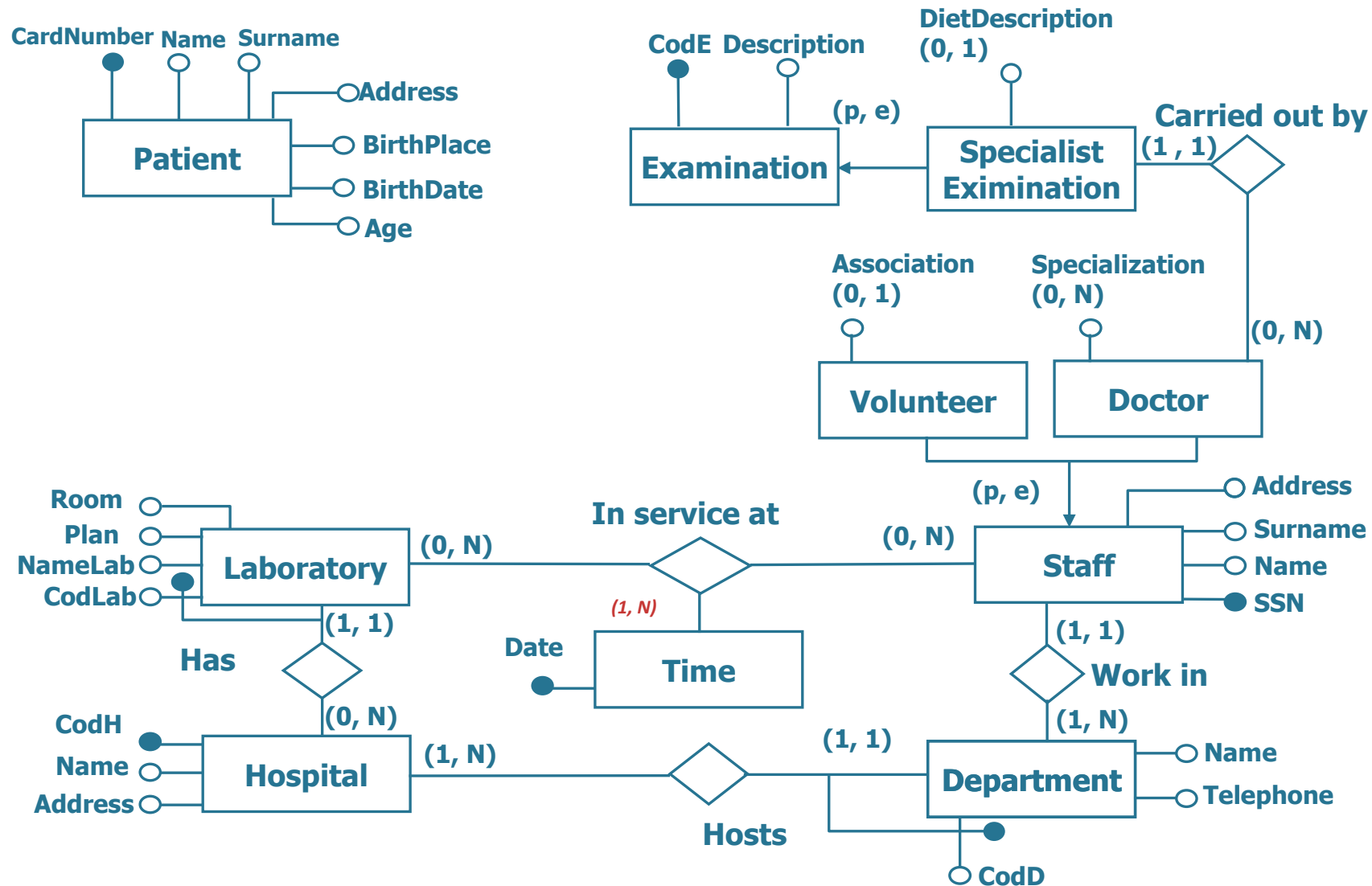
# Cardinality of the *In service at* relationship



# Cardinality of the *In service at* relationship

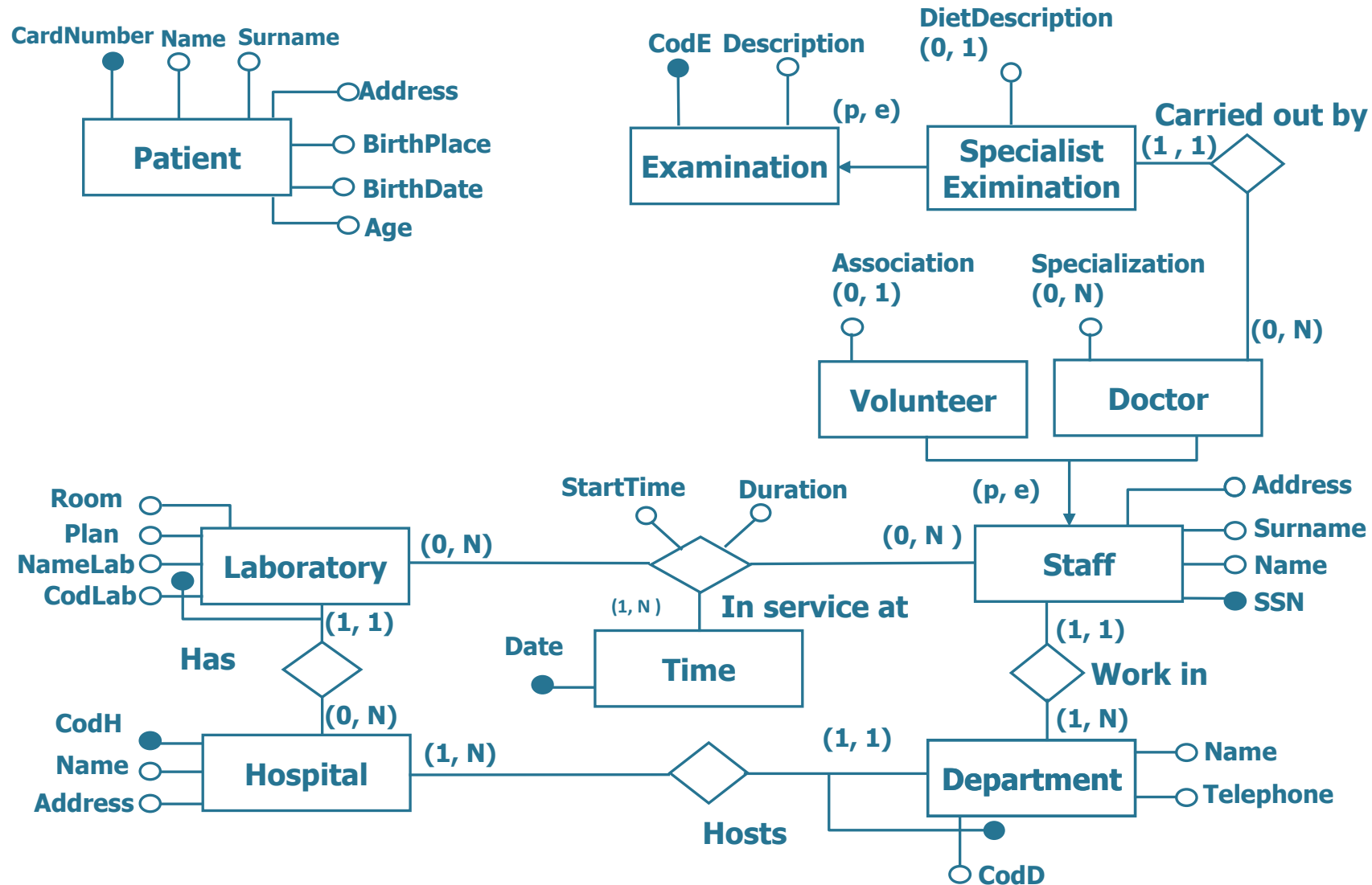


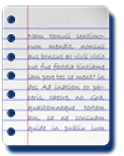
# Cardinality of the *In service at* relationship





# Cardinality of the *In service at* relationship

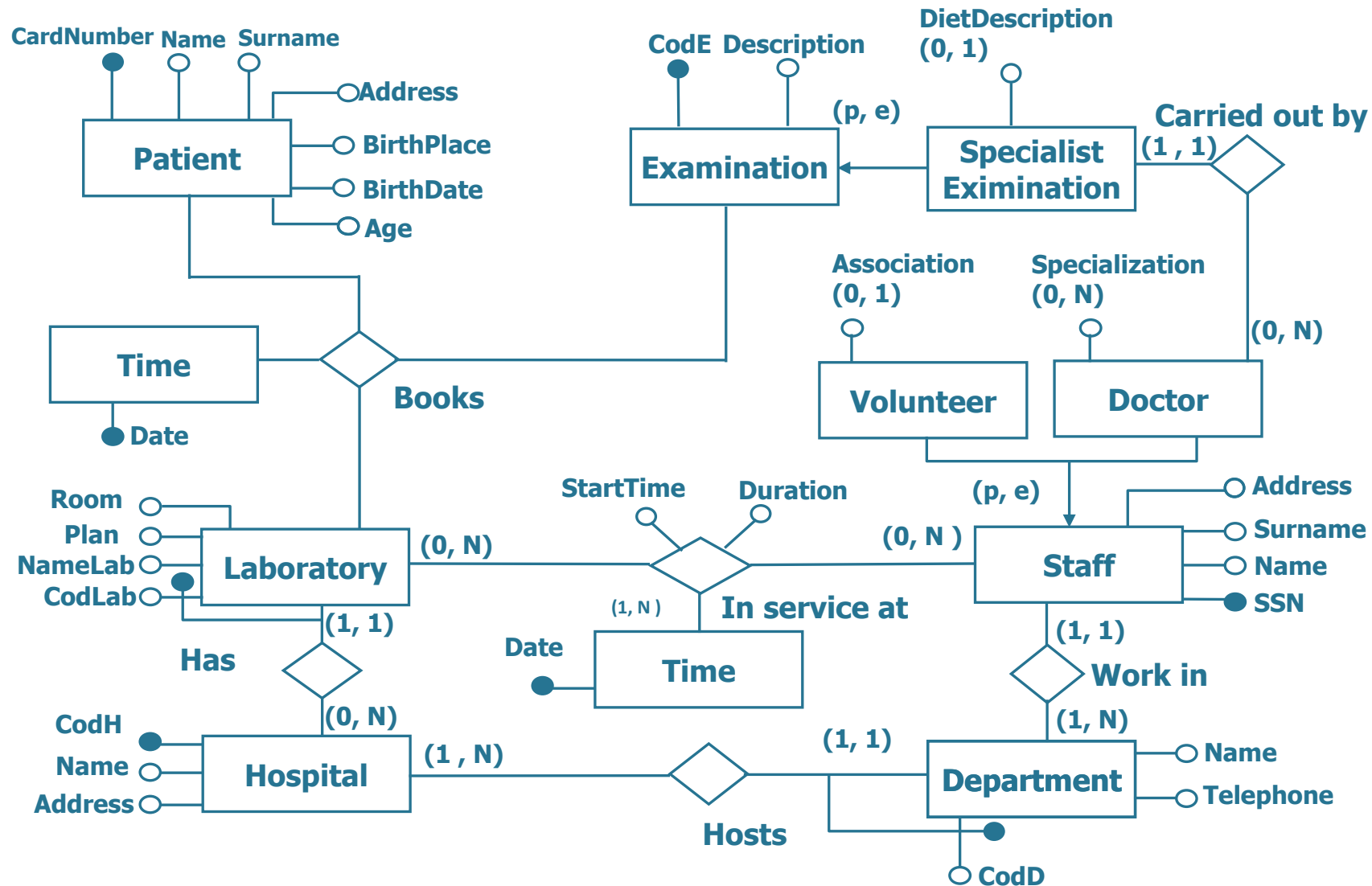




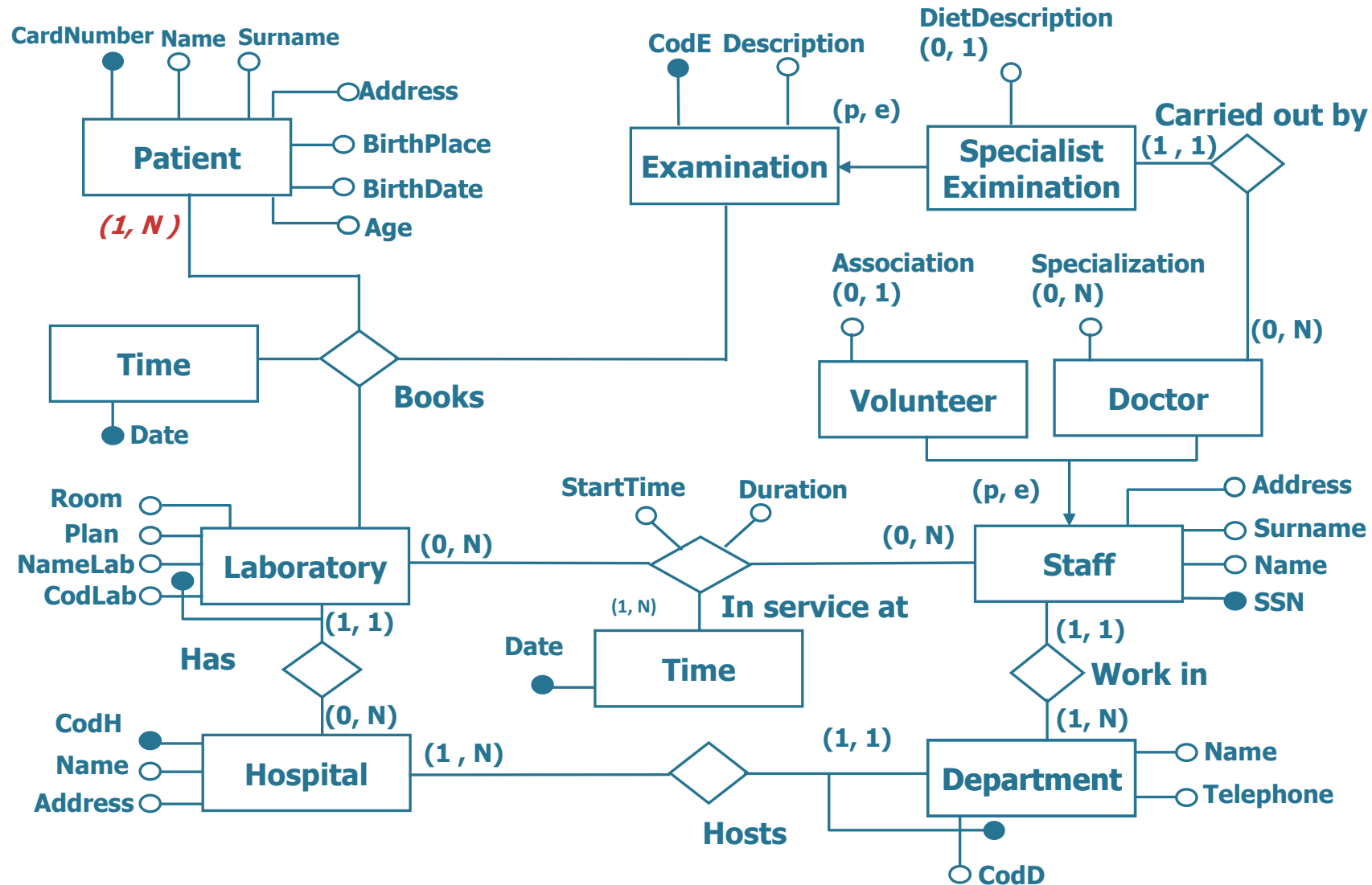
# Representation of the reservation

- Each exam requires a reservation.
- For each exam reservation made by a patient, the date and time of the exam, the laboratory where it is performed, the cost of the ticket and the information about the exam being urgent or not are stored.
- Please note that each patient can make multiple reservations for the same exam on different dates and the same exam cannot be repeated on the same day by the same patient, even in different laboratories.

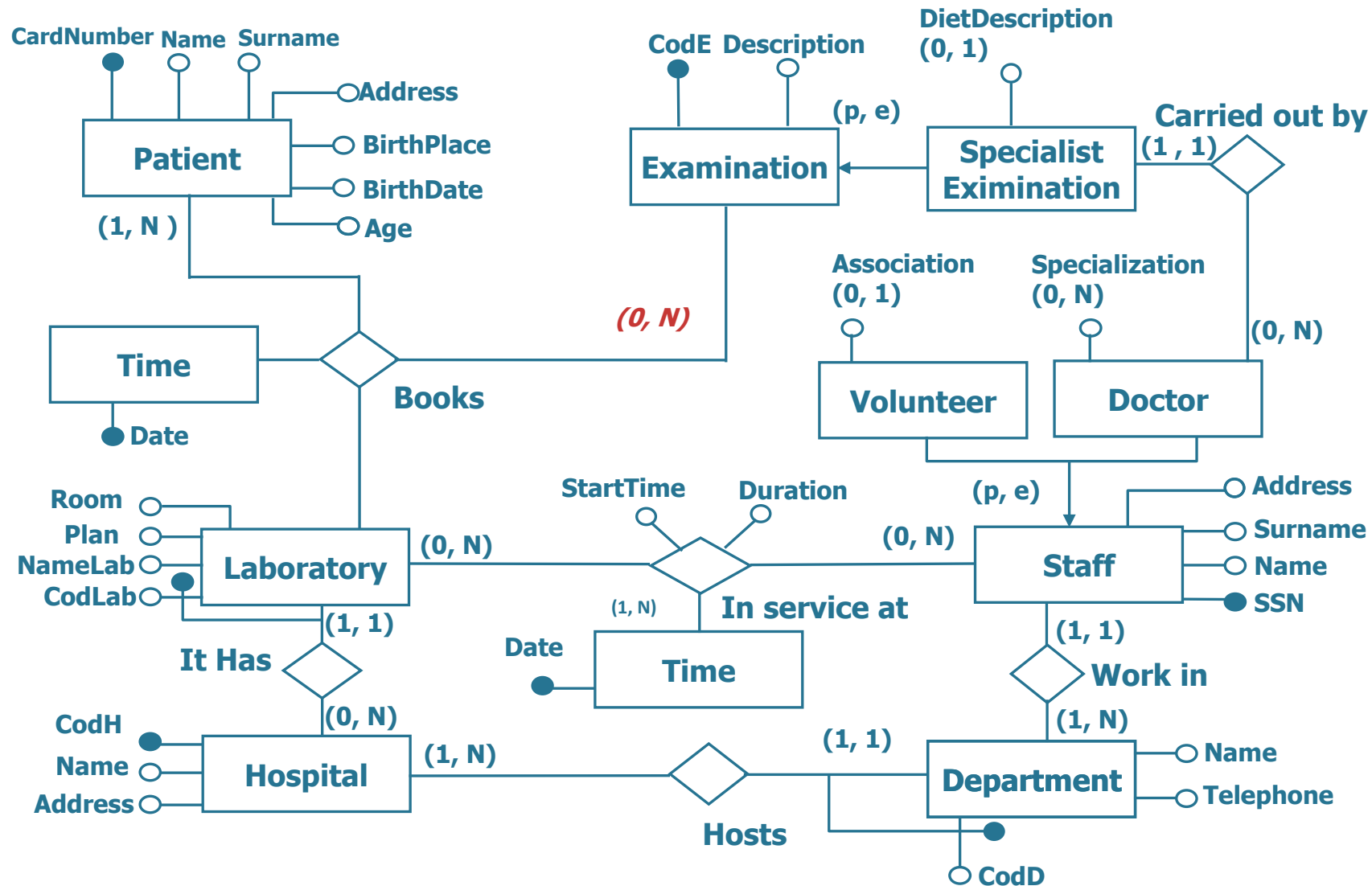
# Representation of the reservation



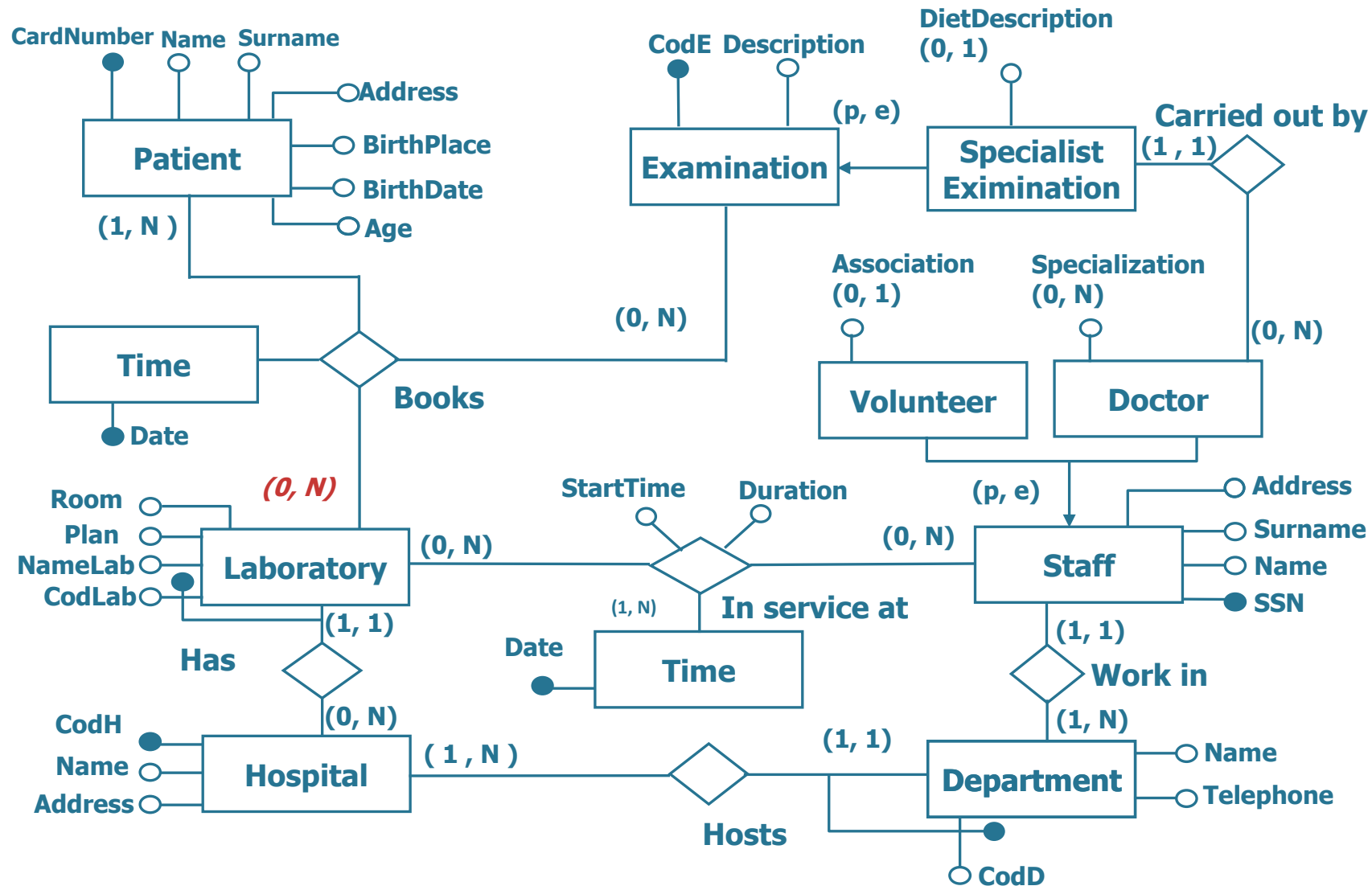
# Cardinality of the *Books* relationship



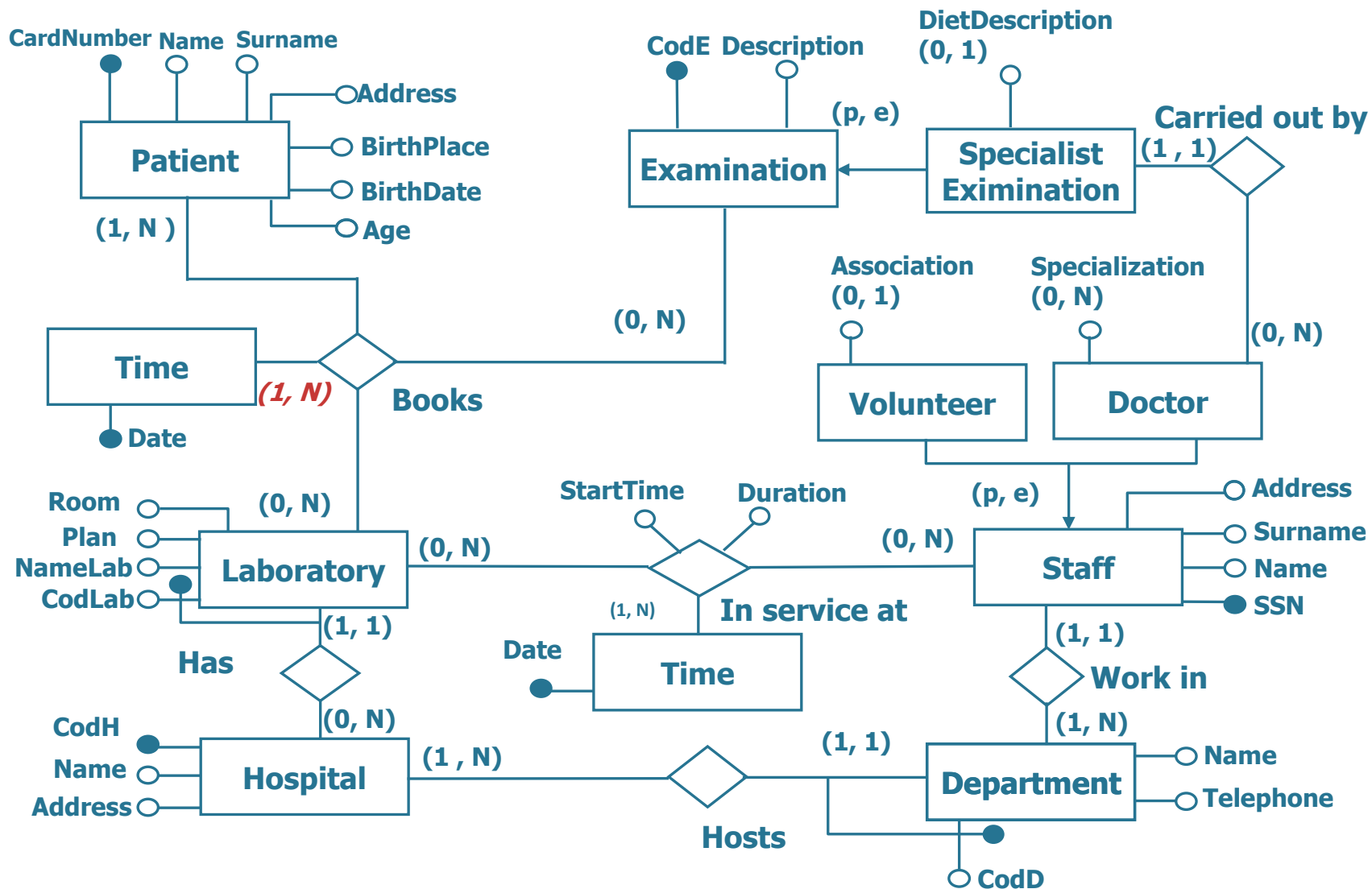
# Cardinality of the *Books* relationship

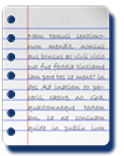


# Cardinality of the *Books* relationship



# Cardinality of the *Books* relationship



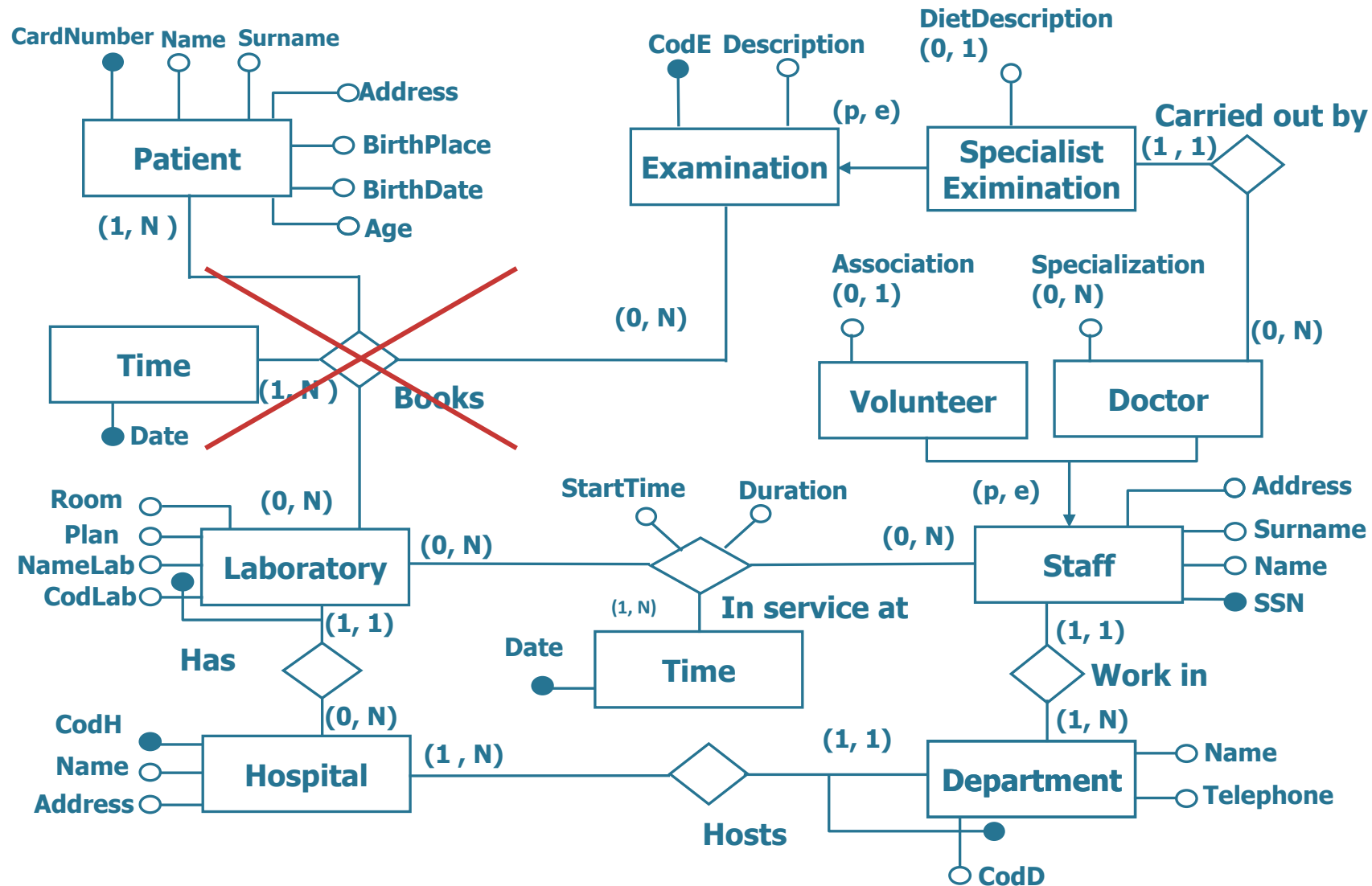


# Reservation constraints

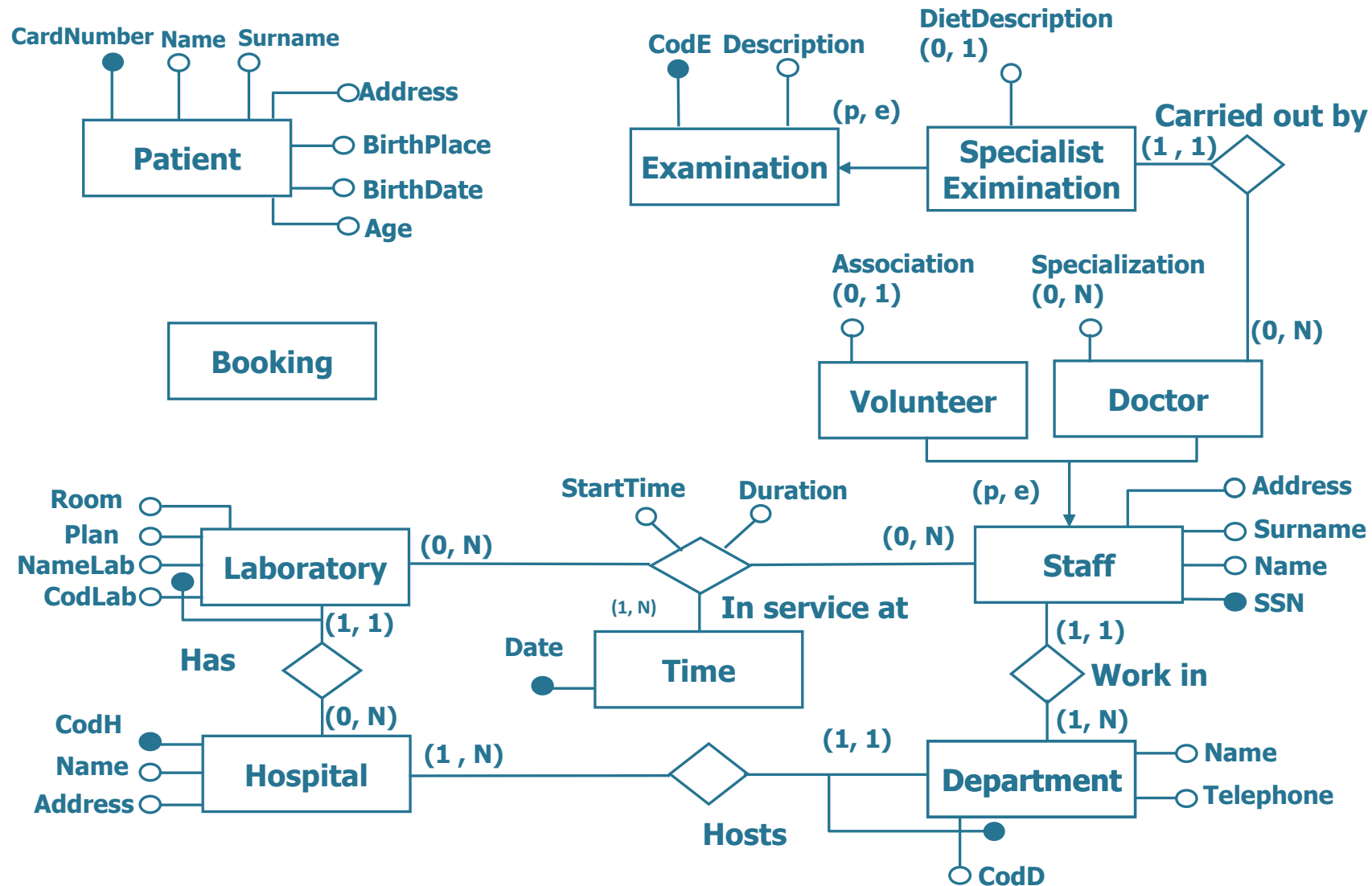
- Each exam requires a reservation.
- For each exam reservation made by a patient, the date and time of the exam, the laboratory where it is performed, the cost of the ticket and the information about the exam being urgent or not are stored.
- *Please note that each patient can make multiple reservations for the same exam on different dates and the same exam cannot be repeated on the same day by the same patient, even in different laboratories.*



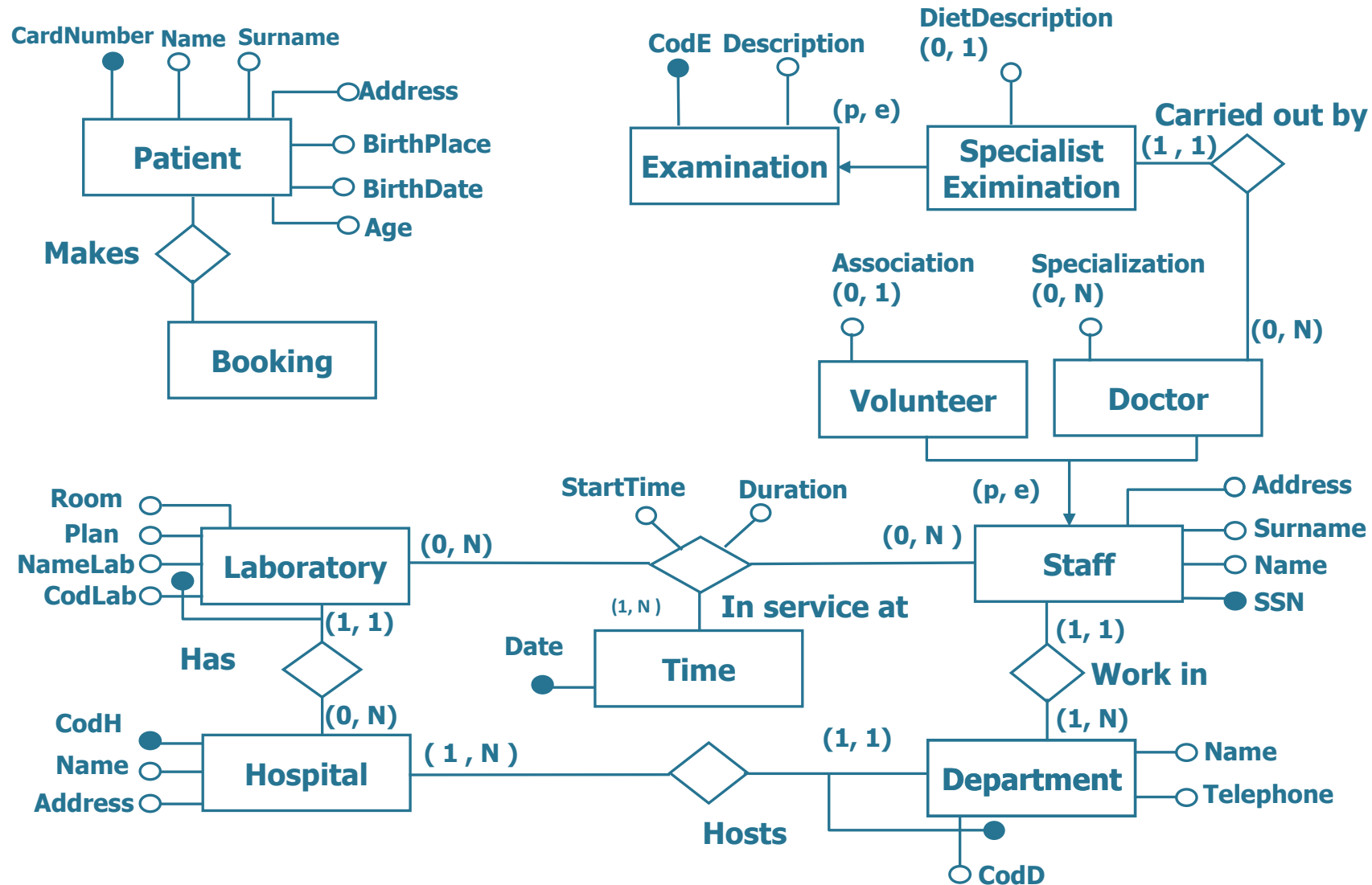
# Representation of the reservation



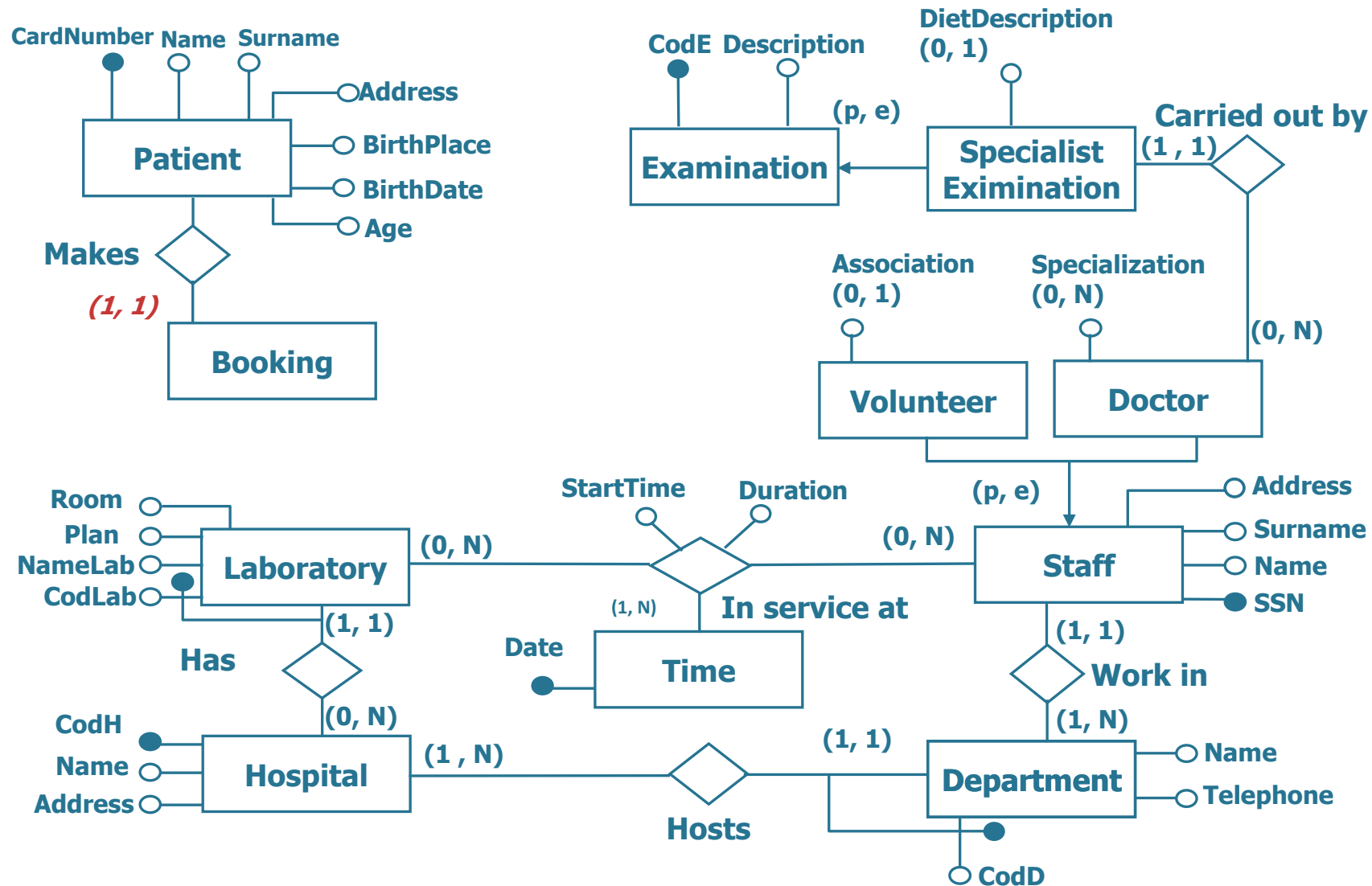
# Introducing the Booking entity



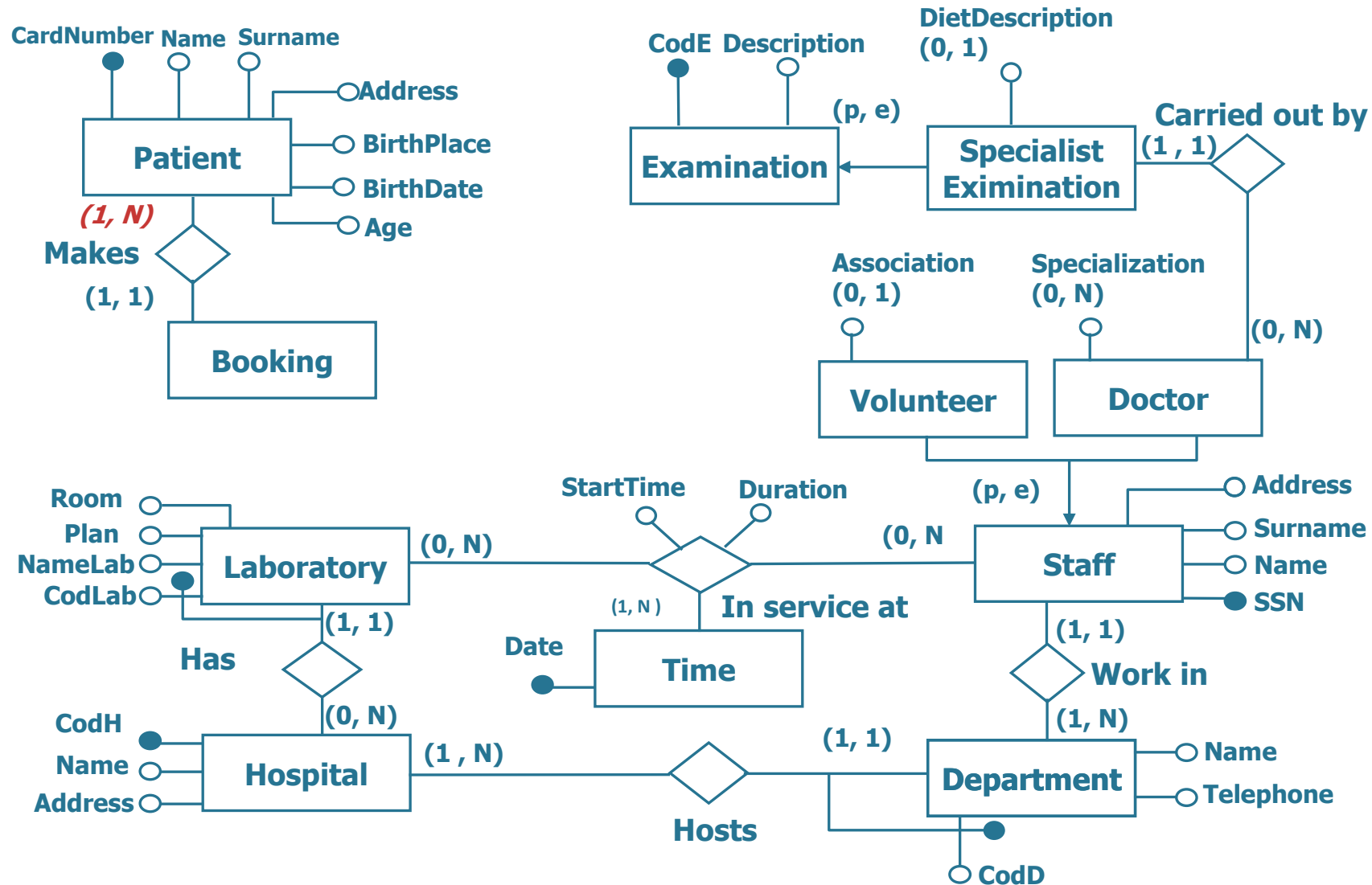
# Relationship between Booking and Patient



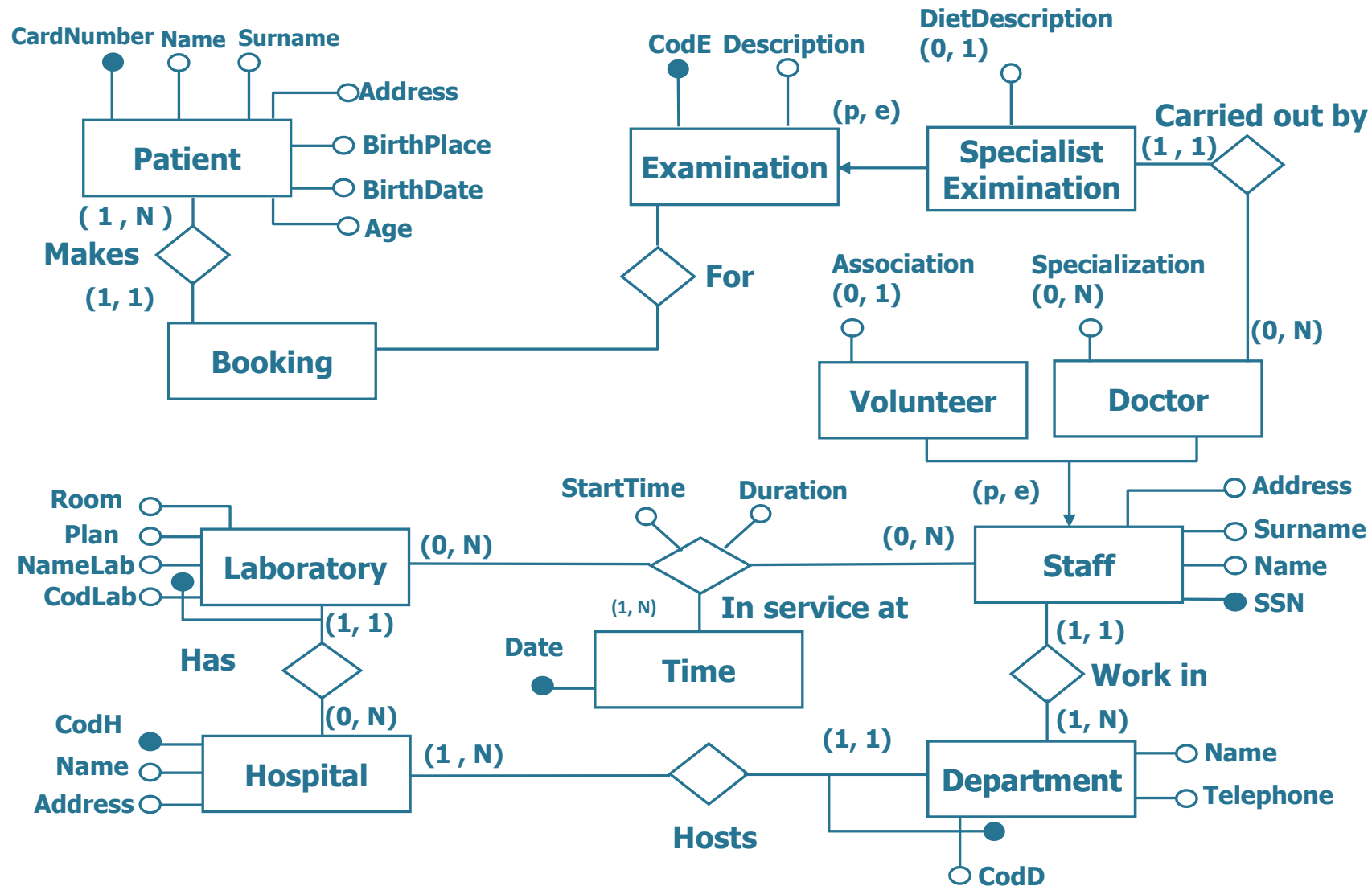
# Cardinality of the *Makes* relationship



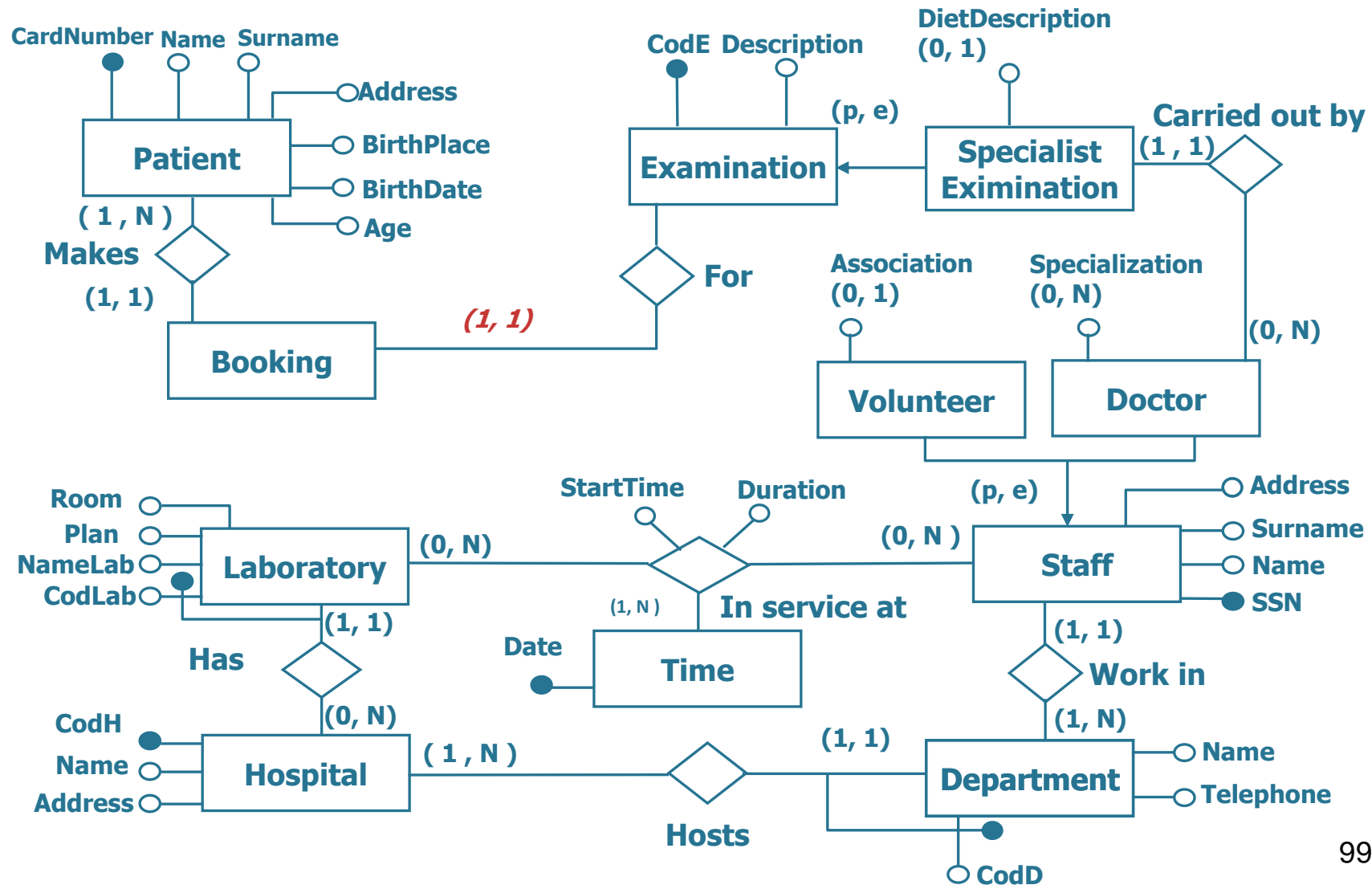
# Cardinality of the *Makes* relationship



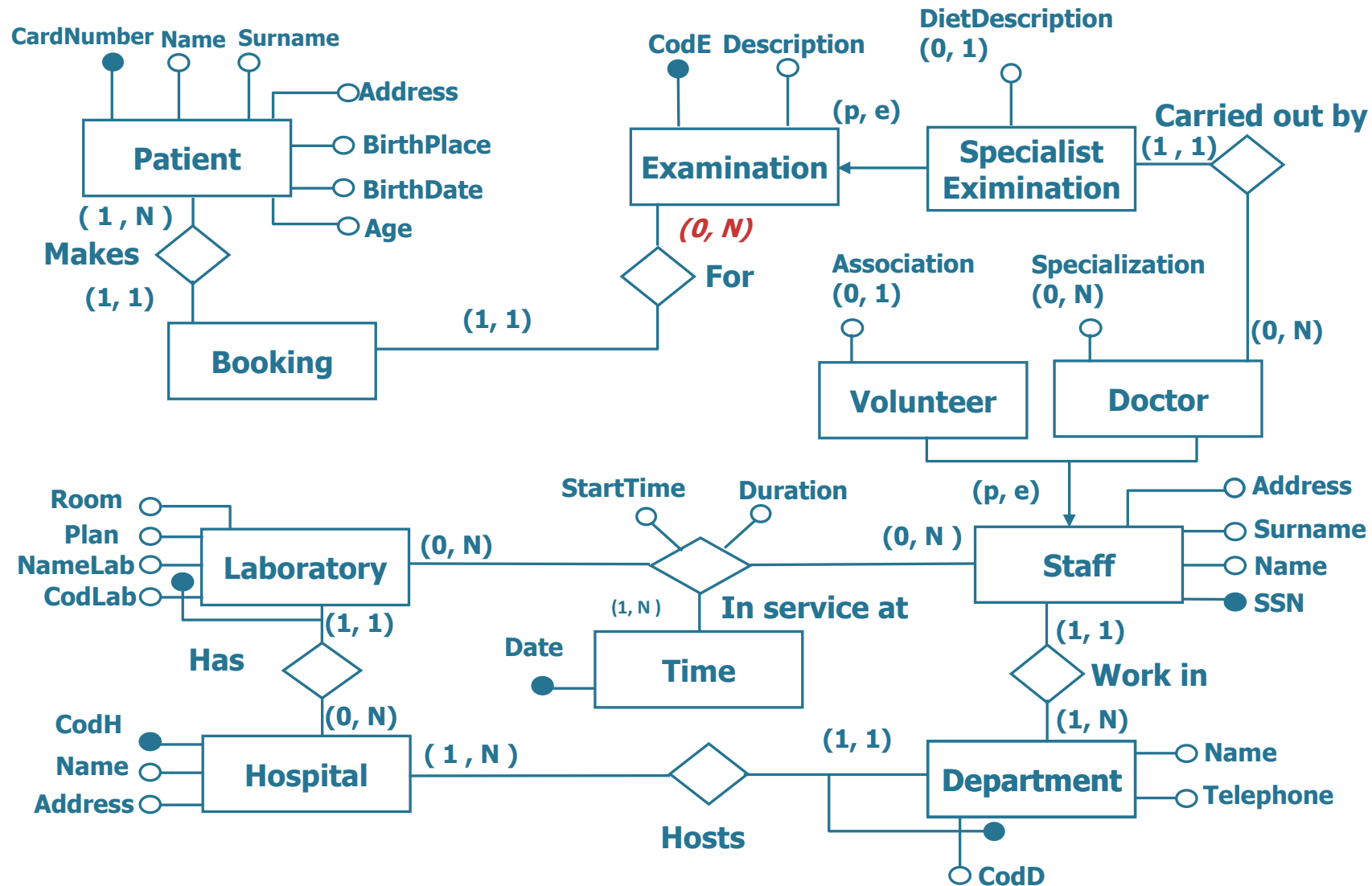
# Relationship between Booking and Patient



# Cardinality of the *For* relationship

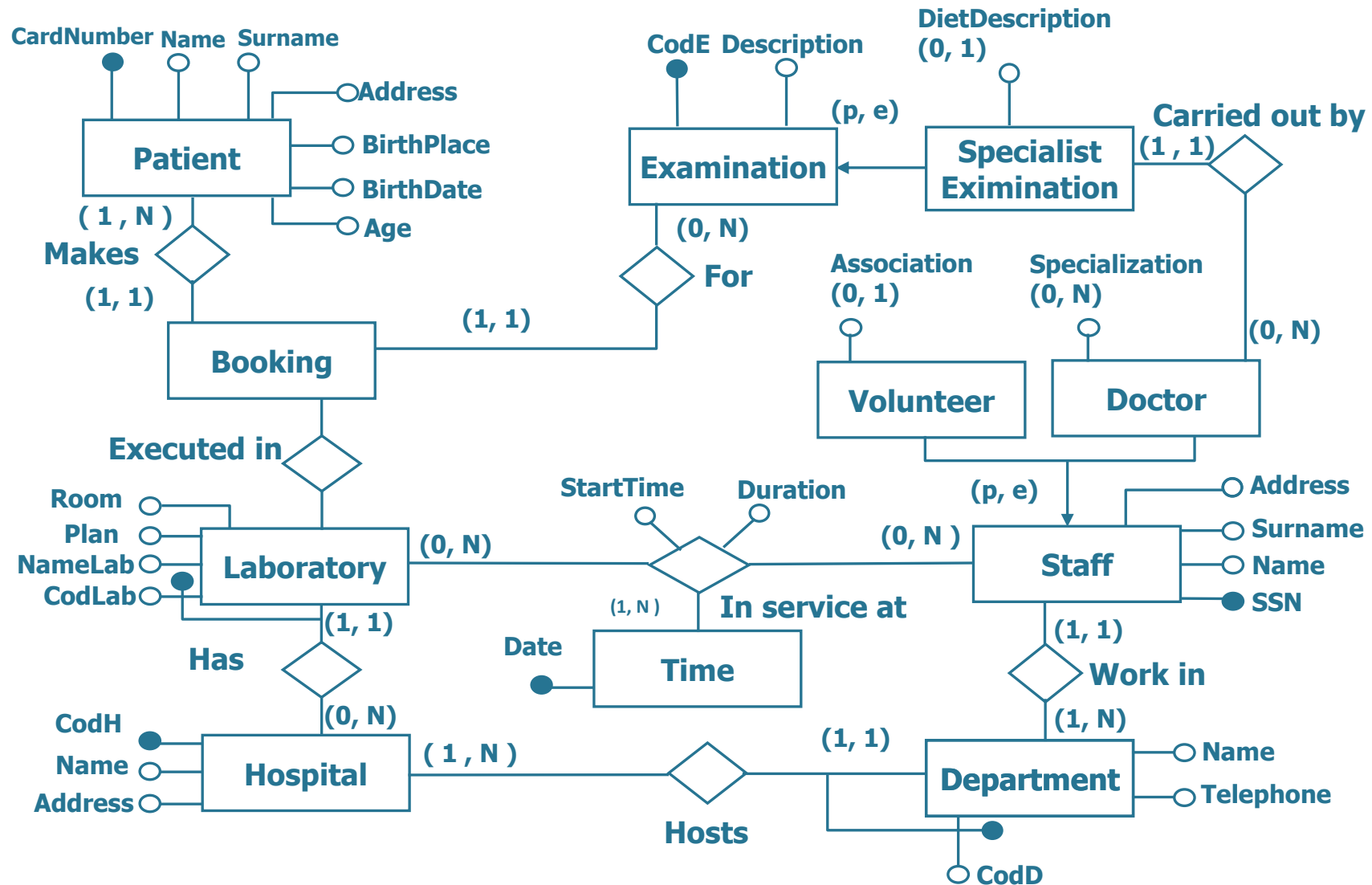


# Cardinality of the *For* relationship

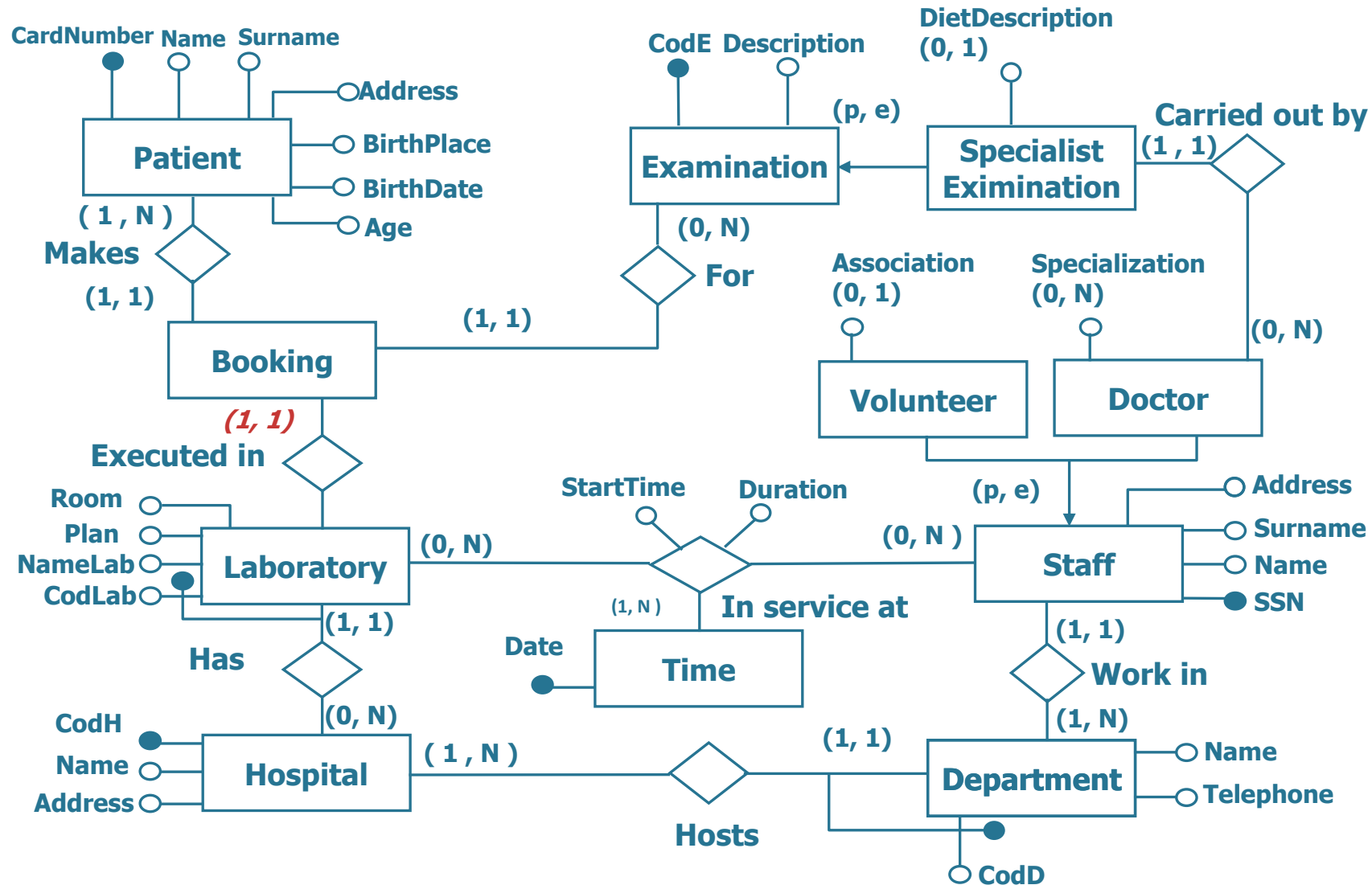




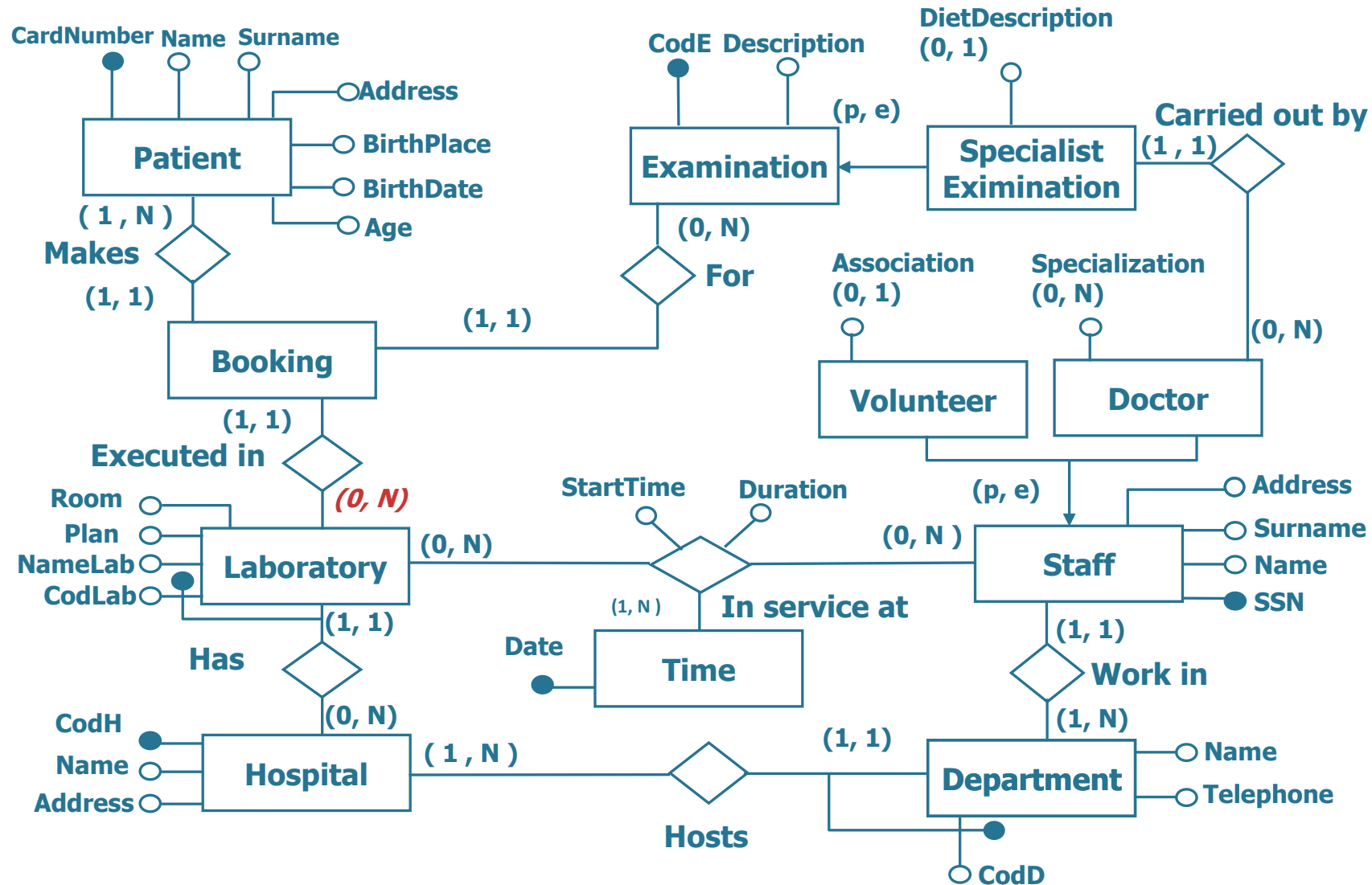
# Relationship between Booking and Laboratory



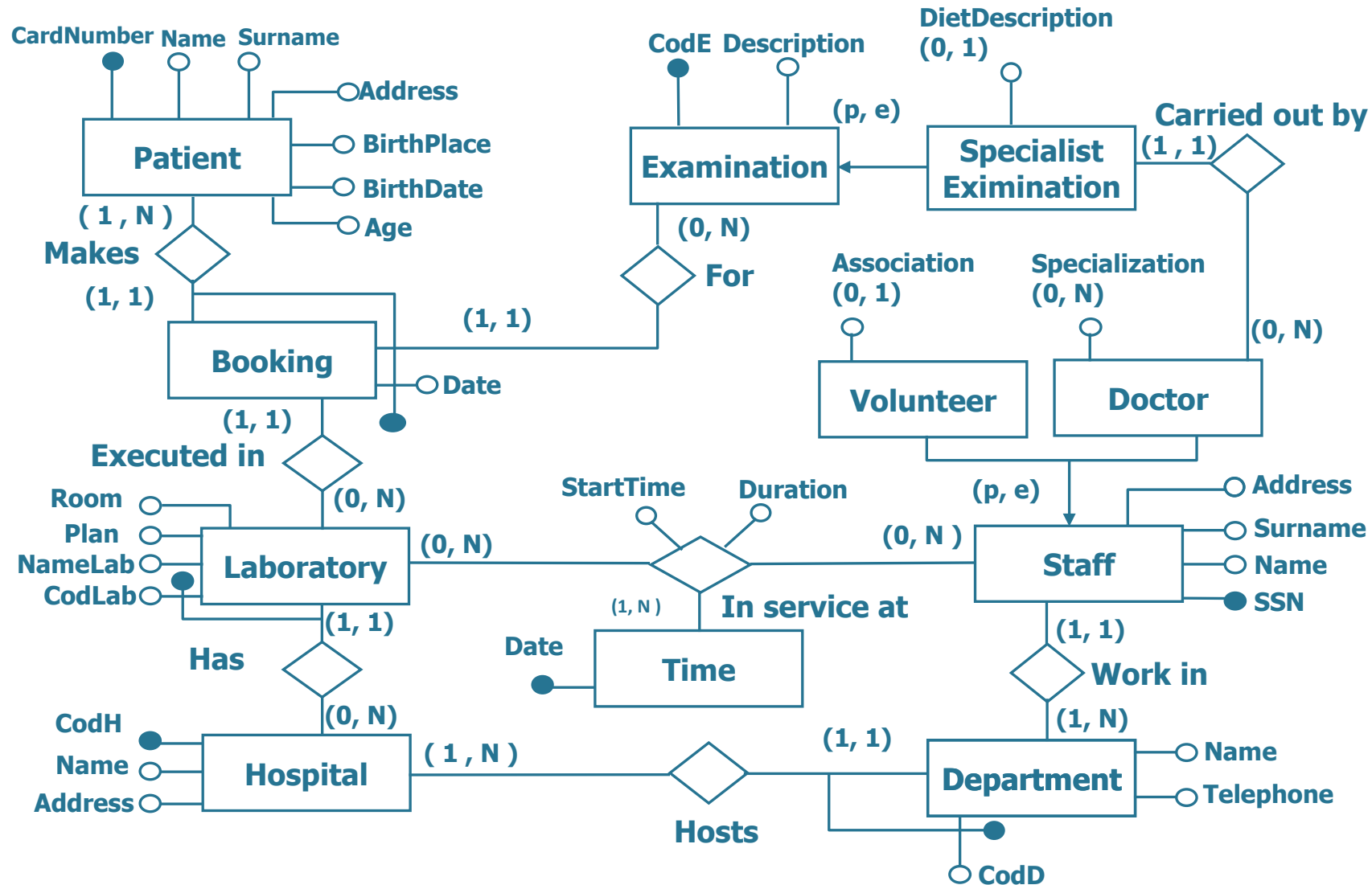
# Cardinality of the *Executed in* relationship



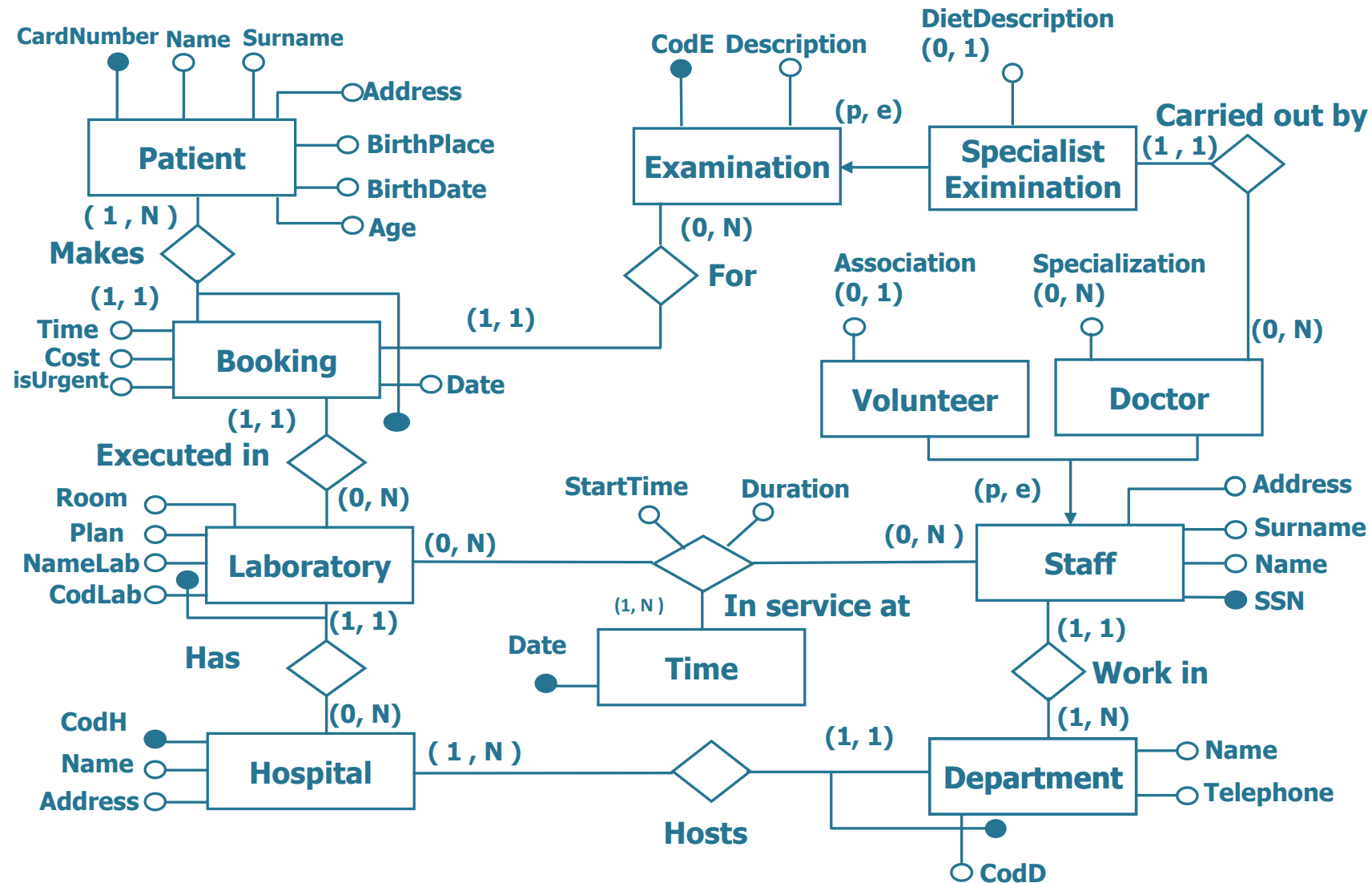
# Cardinality of the *Executed in* relationship

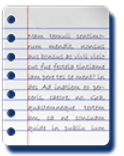


# Booking entity identifier



# Refinement of the Booking entity

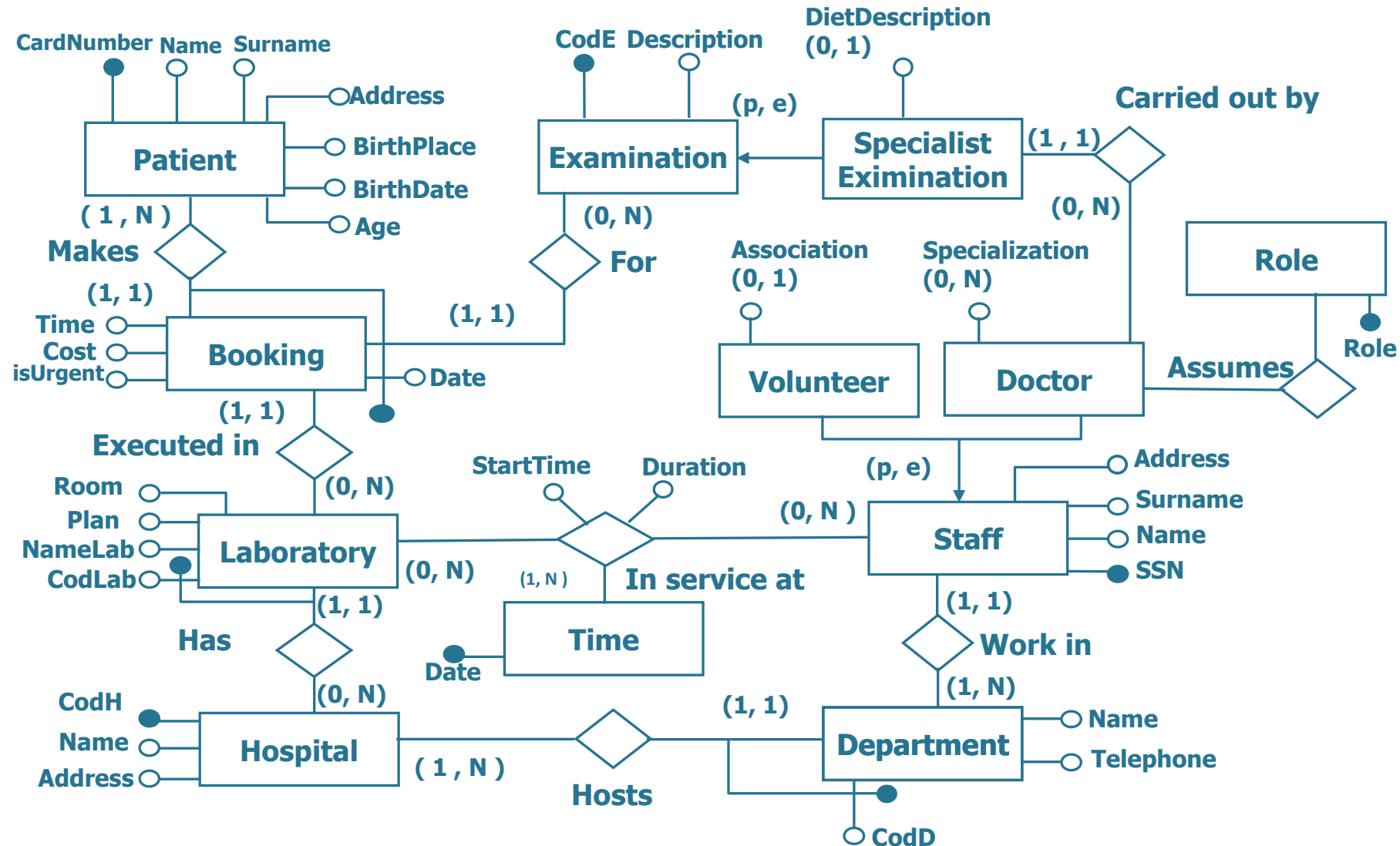




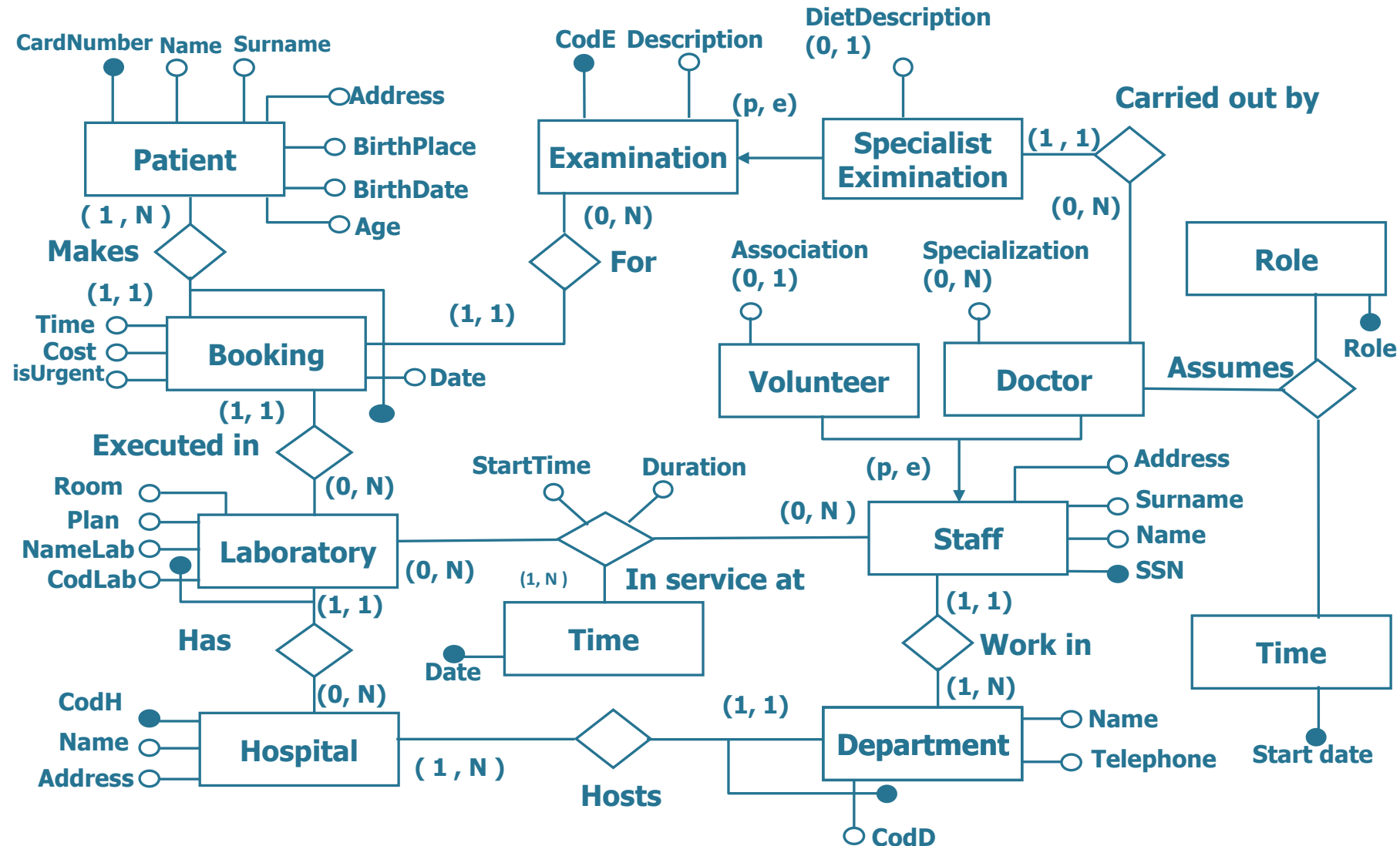
# Relationship between Doctor and Role

- Each doctor can take on different roles during his/her career (e.g. assistant, head physician, etc.).  
We want to keep track of the roles each doctor has taken on during his/her career and the related time periods (start date, end date).  
Keep in mind that each doctor cannot take on more than one role at the same time, but he/she can take on the same role in different time periods.

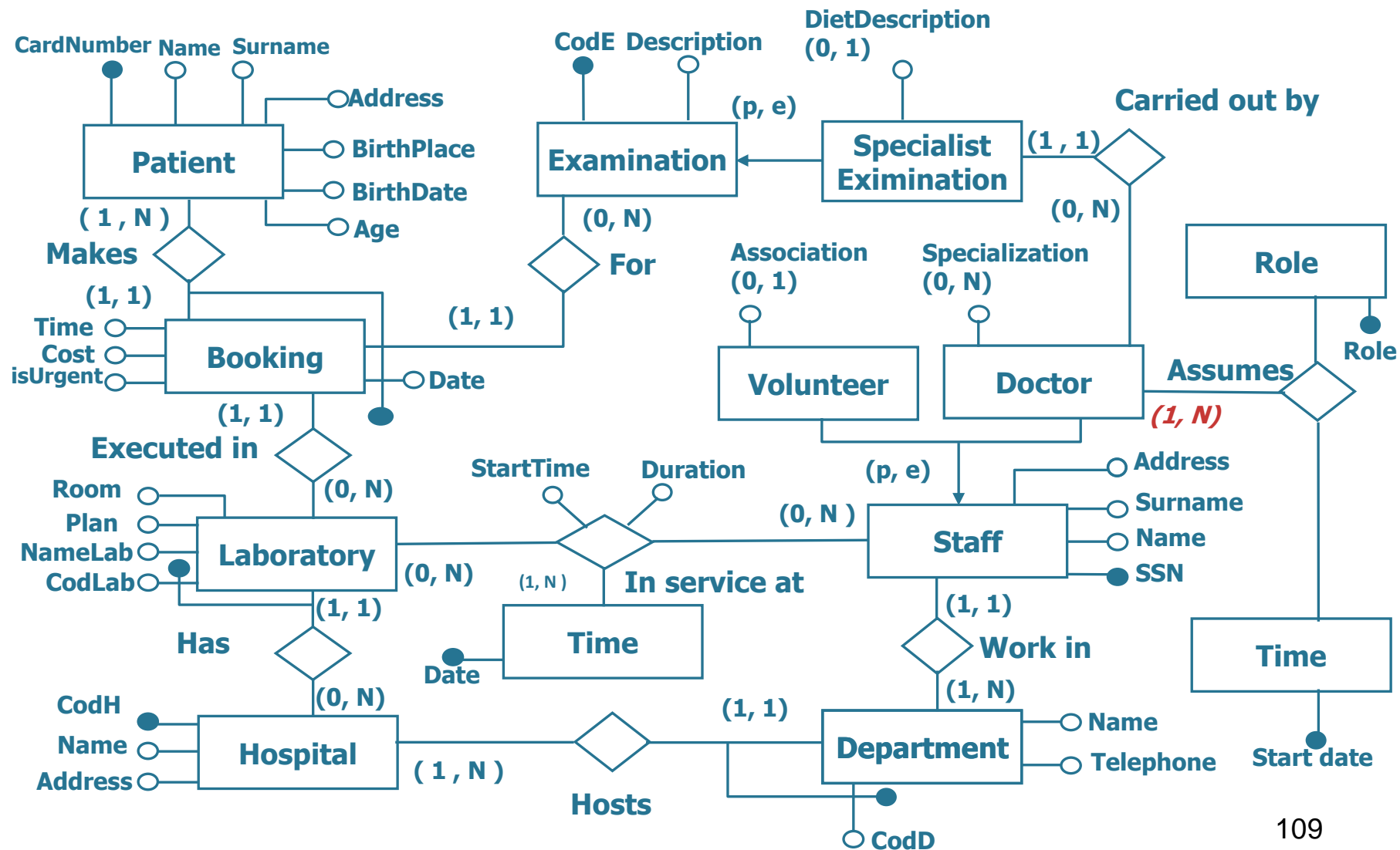
# Relationship between Doctor and Role



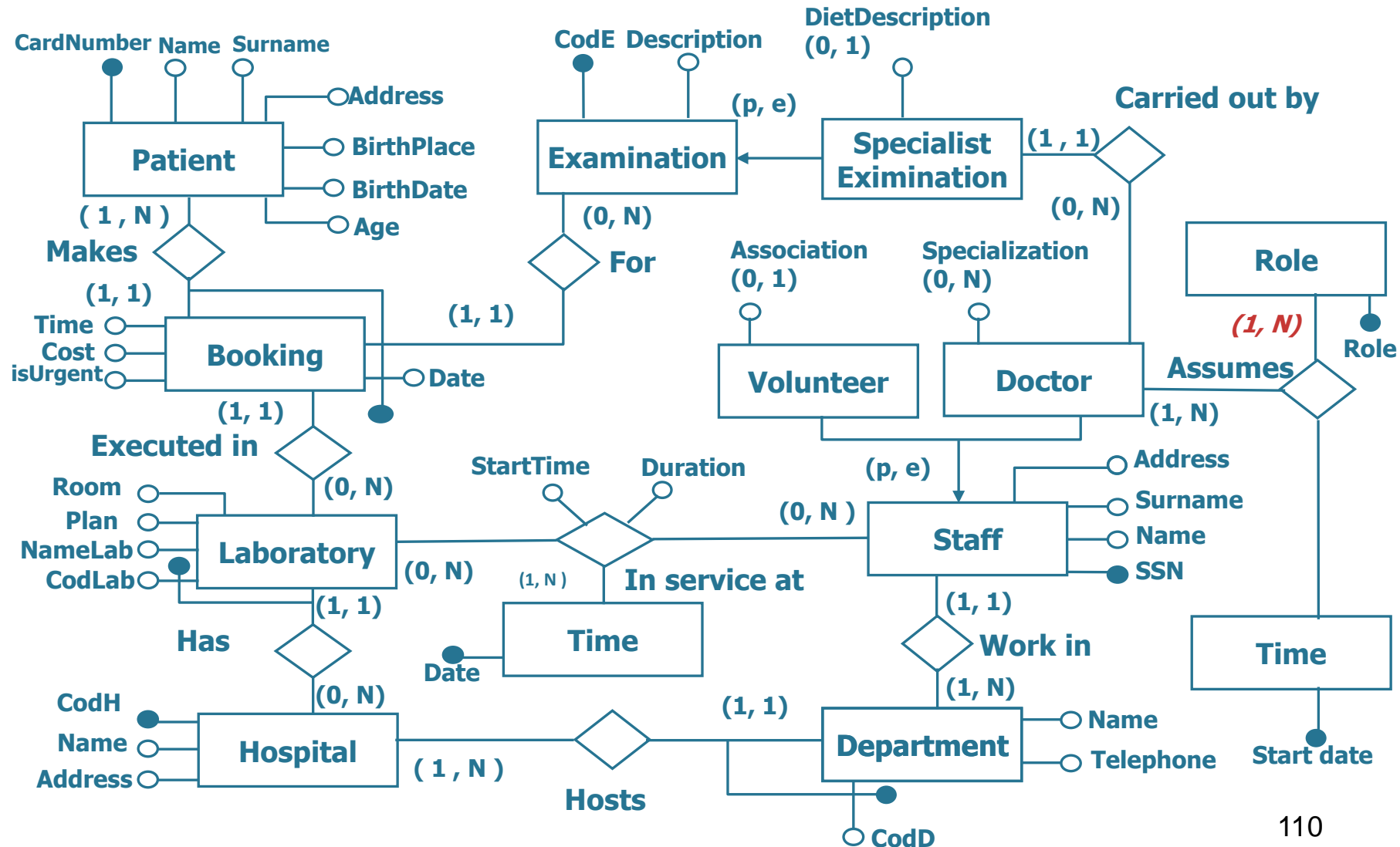
# Relationship between Doctor and Role



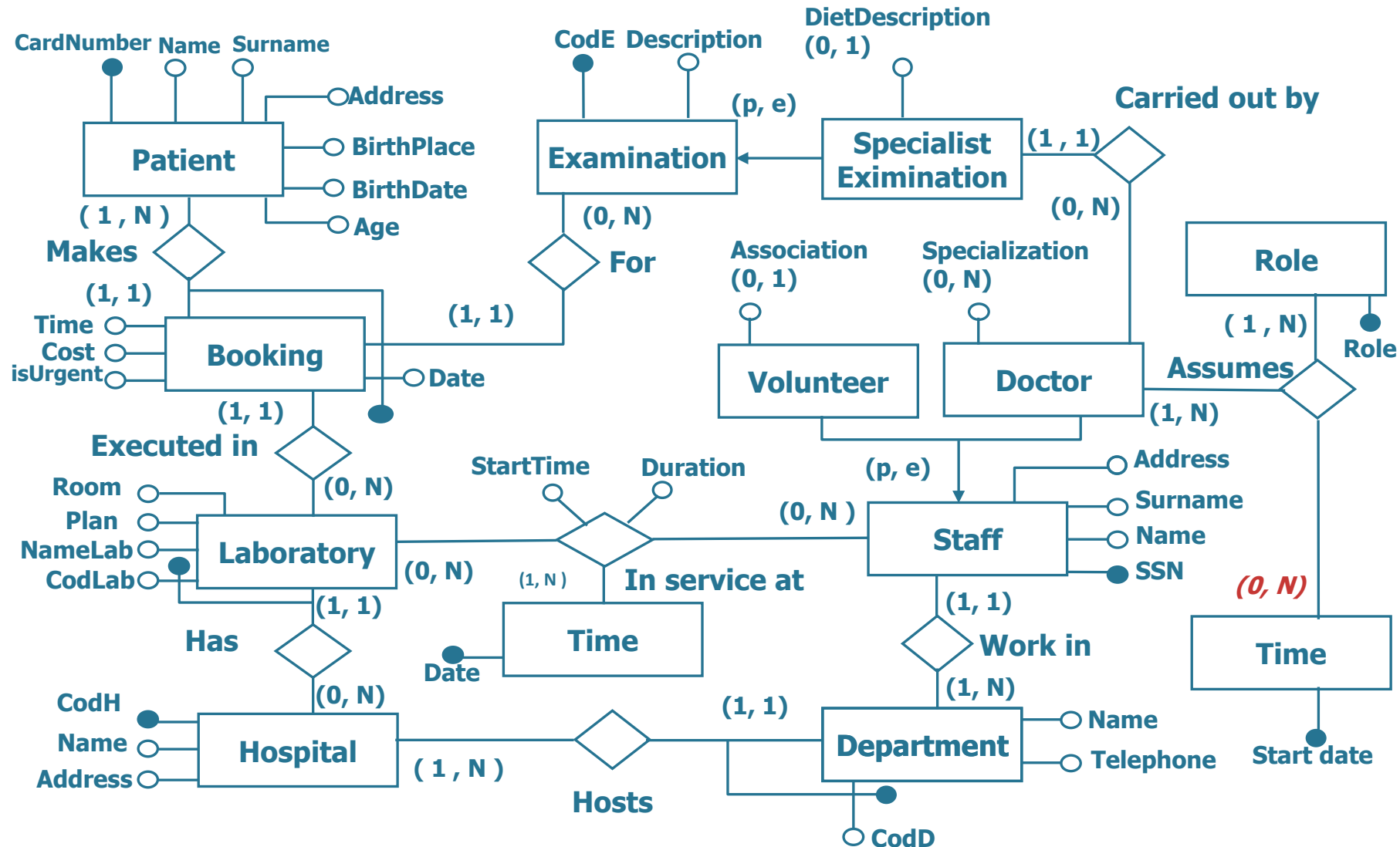




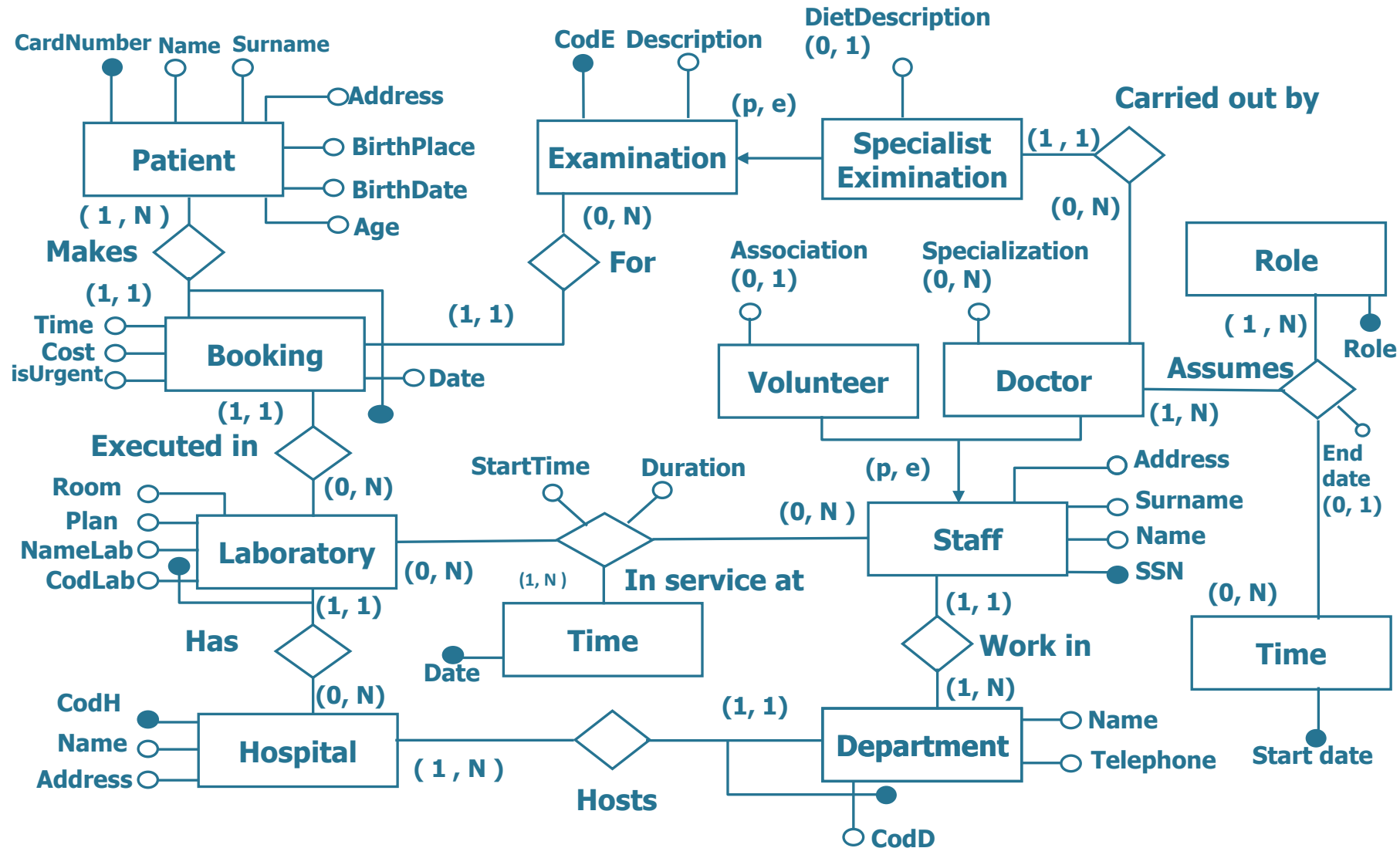
# Cardinality of the *Assumes* relationship

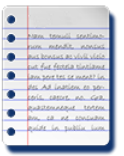


# Cardinality of the *Assumes* relationship



# Cardinality of the *Assumes* relationship





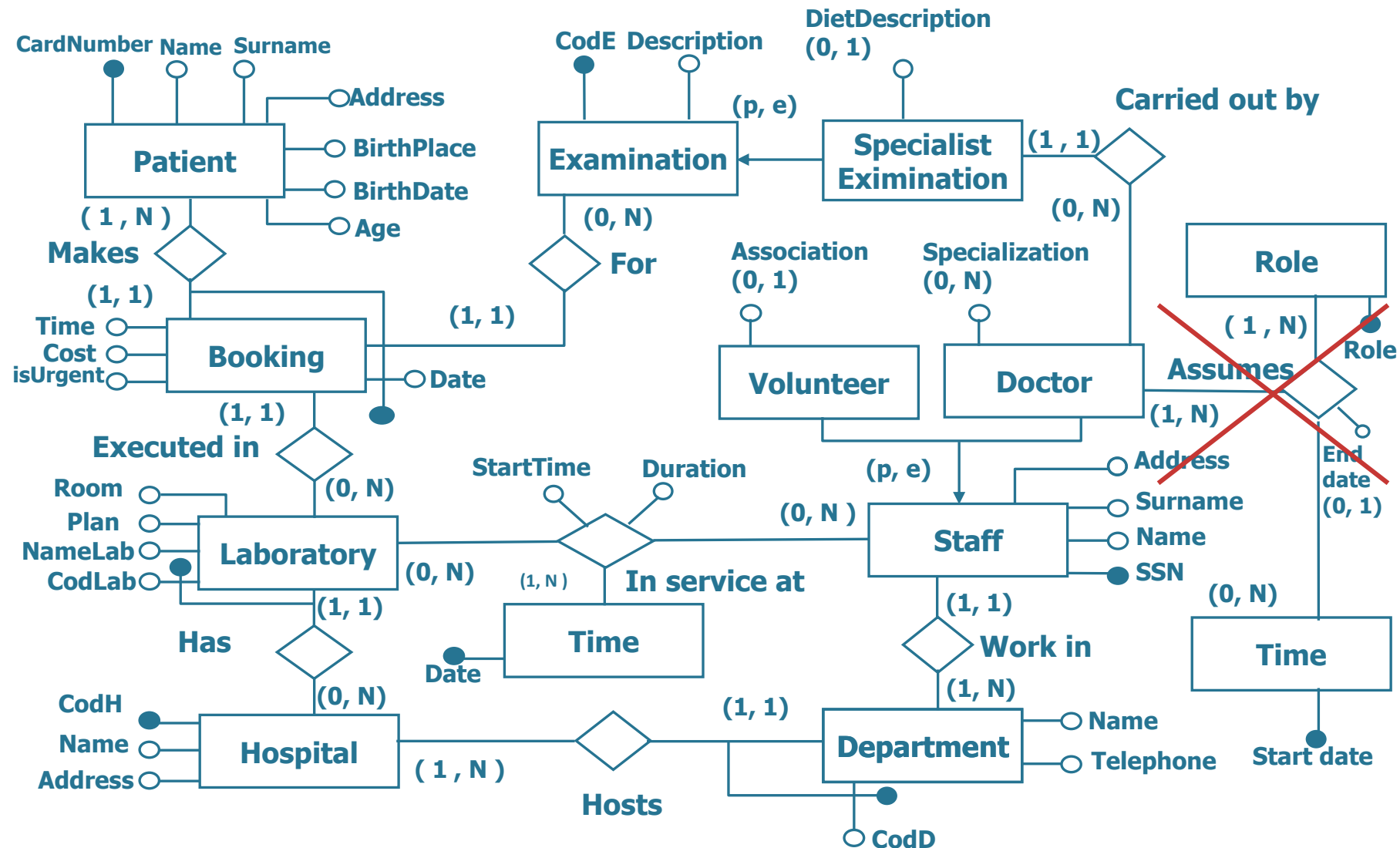
# Constraints on the *Assumes* relationship

- Each doctor can take on different roles during his/her career (e.g. assistant, head physician, etc.).

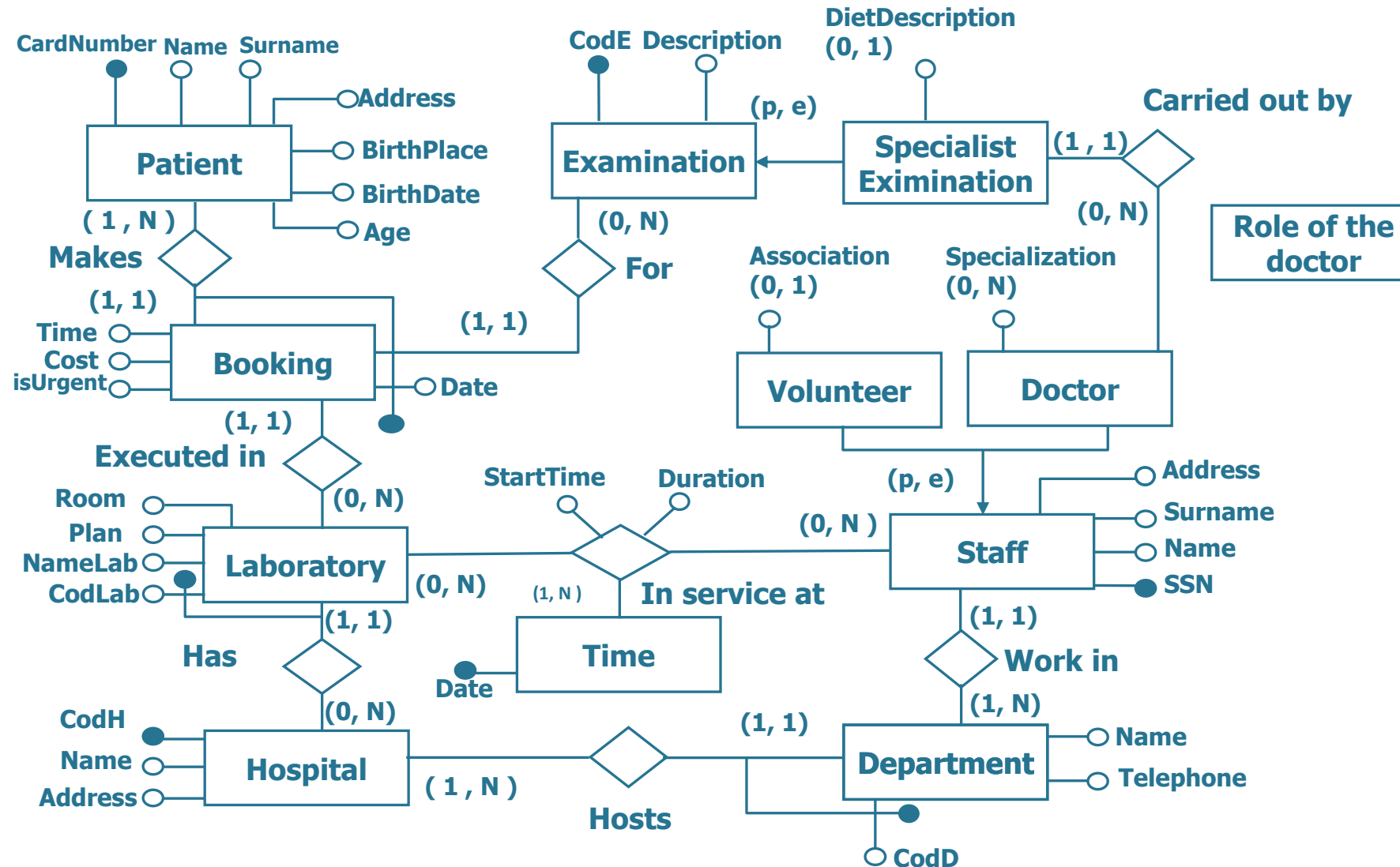
We want to keep track of the roles each doctor has taken on during his/her career and the related time periods (start date, end date).

*Keep in mind that each doctor cannot take on more than one role at the same time, but he/she can take on the same role in different time periods.*

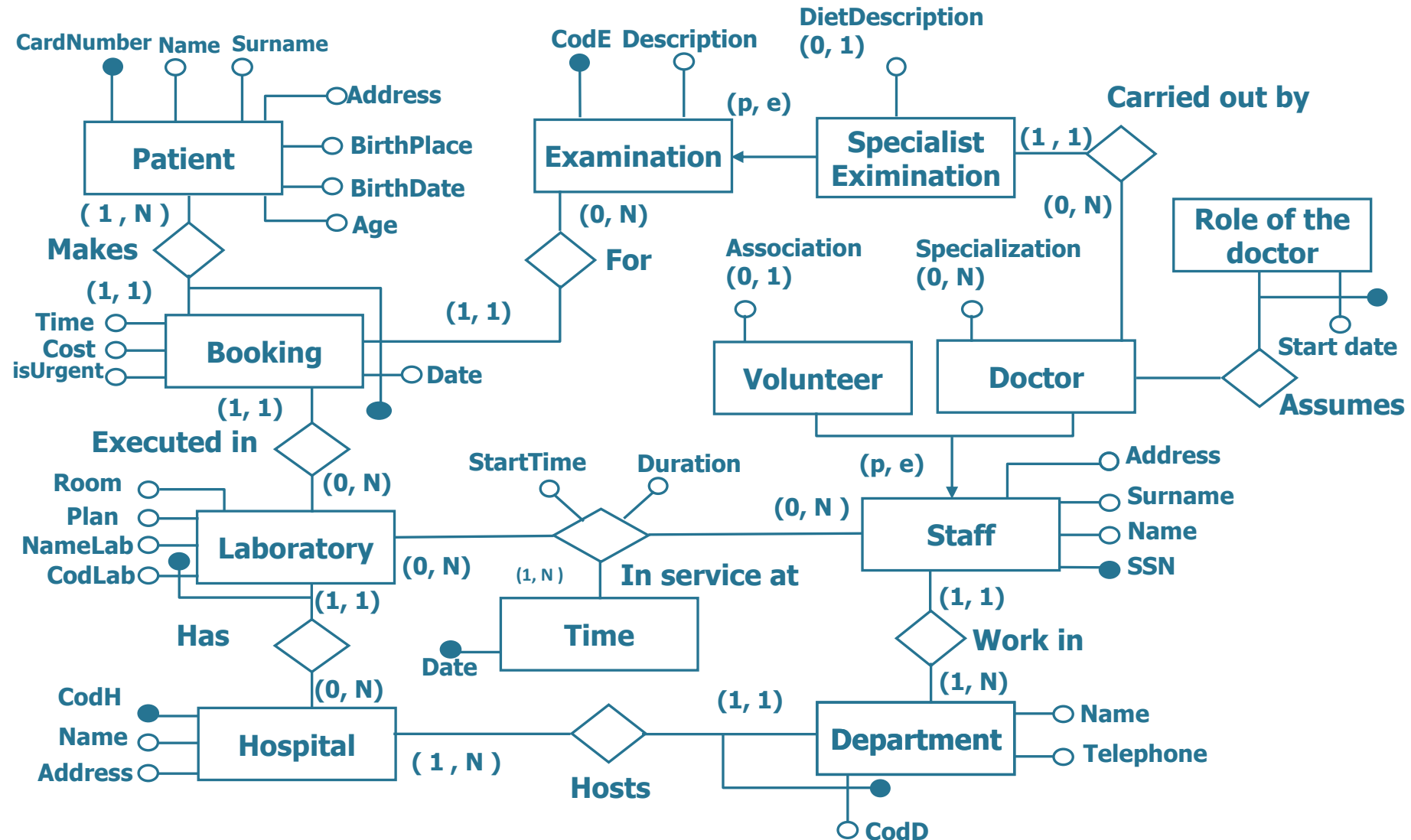
# Historicization of the role of the doctor



# Historicization of the role of the doctor

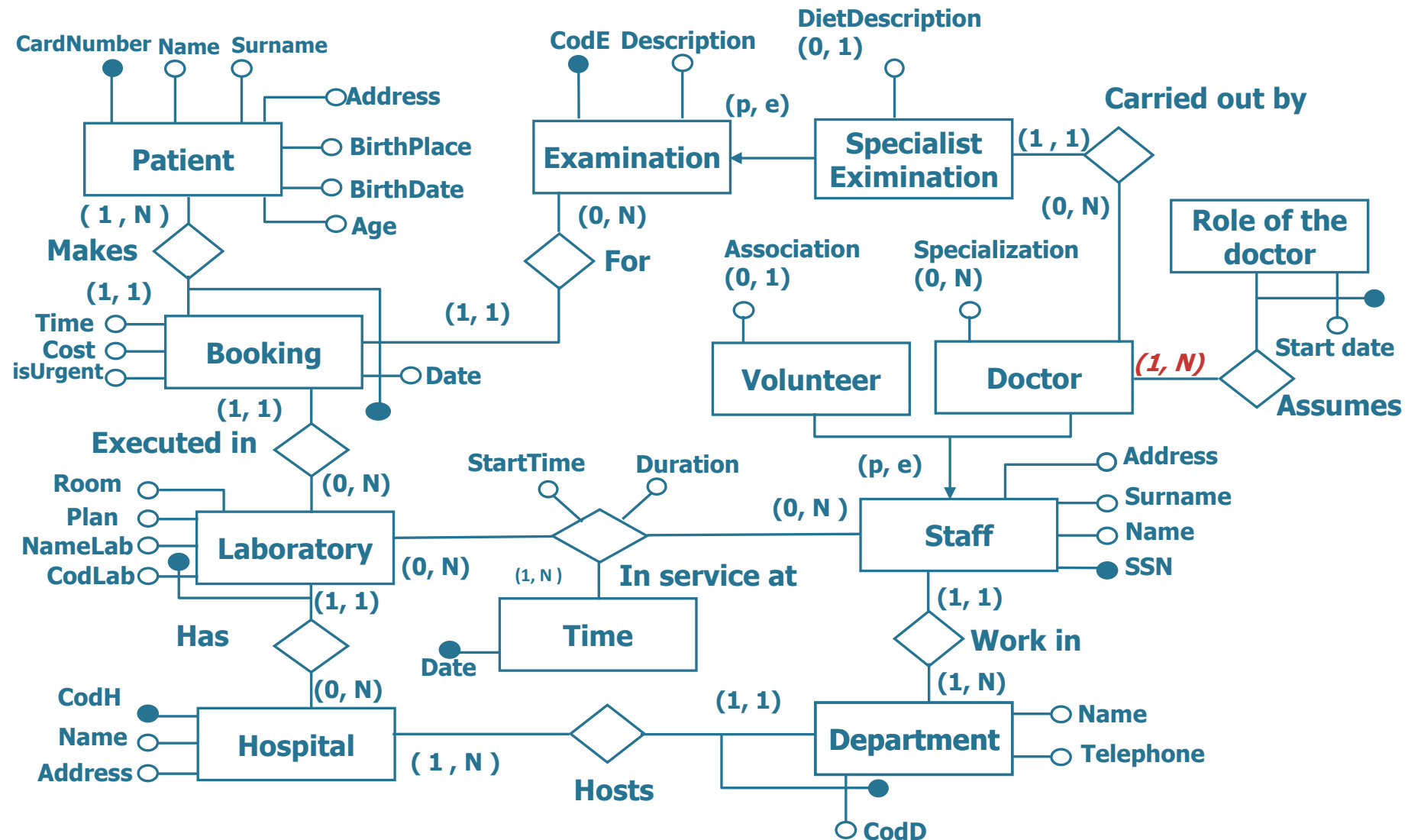


# Historicization of the role of the doctor

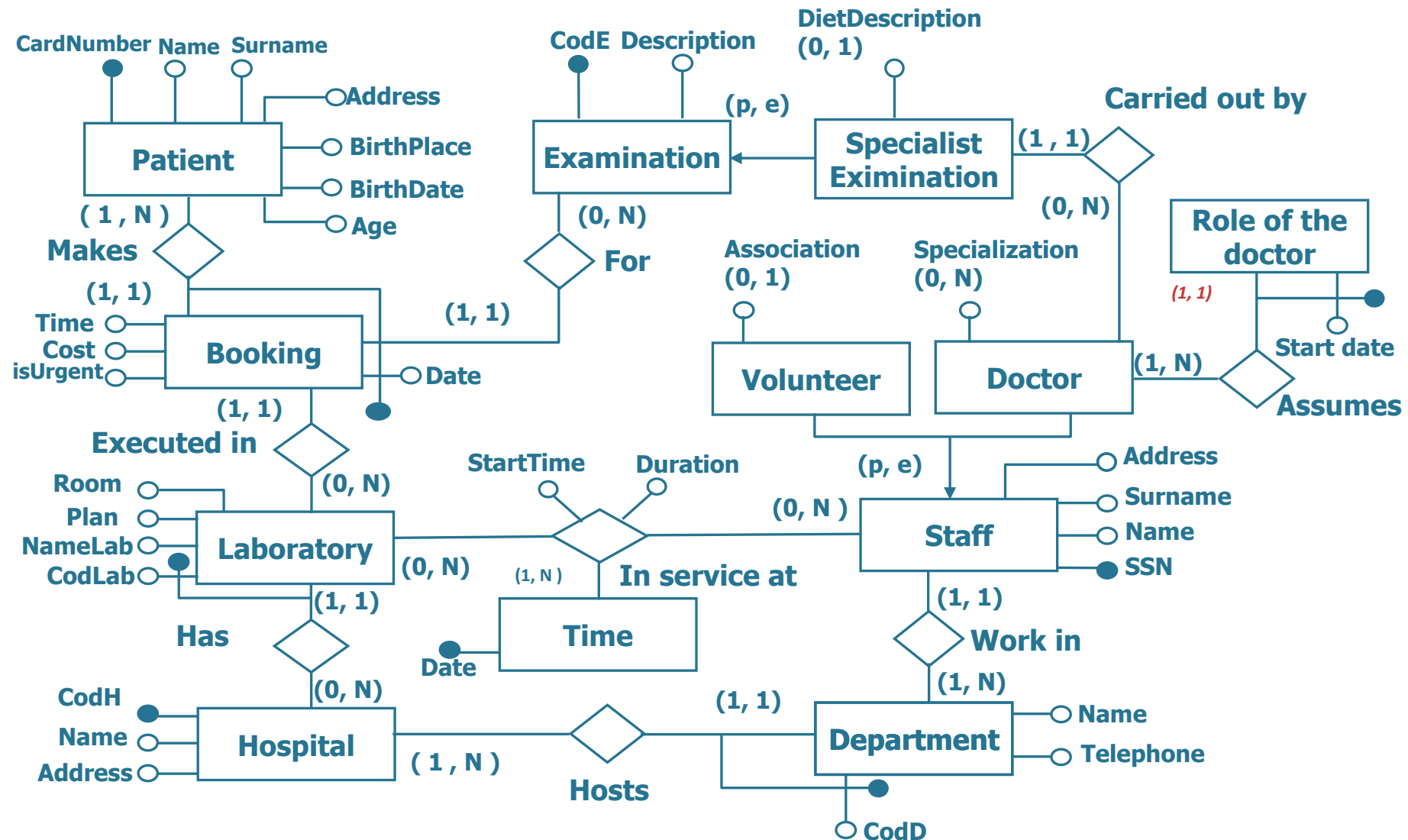




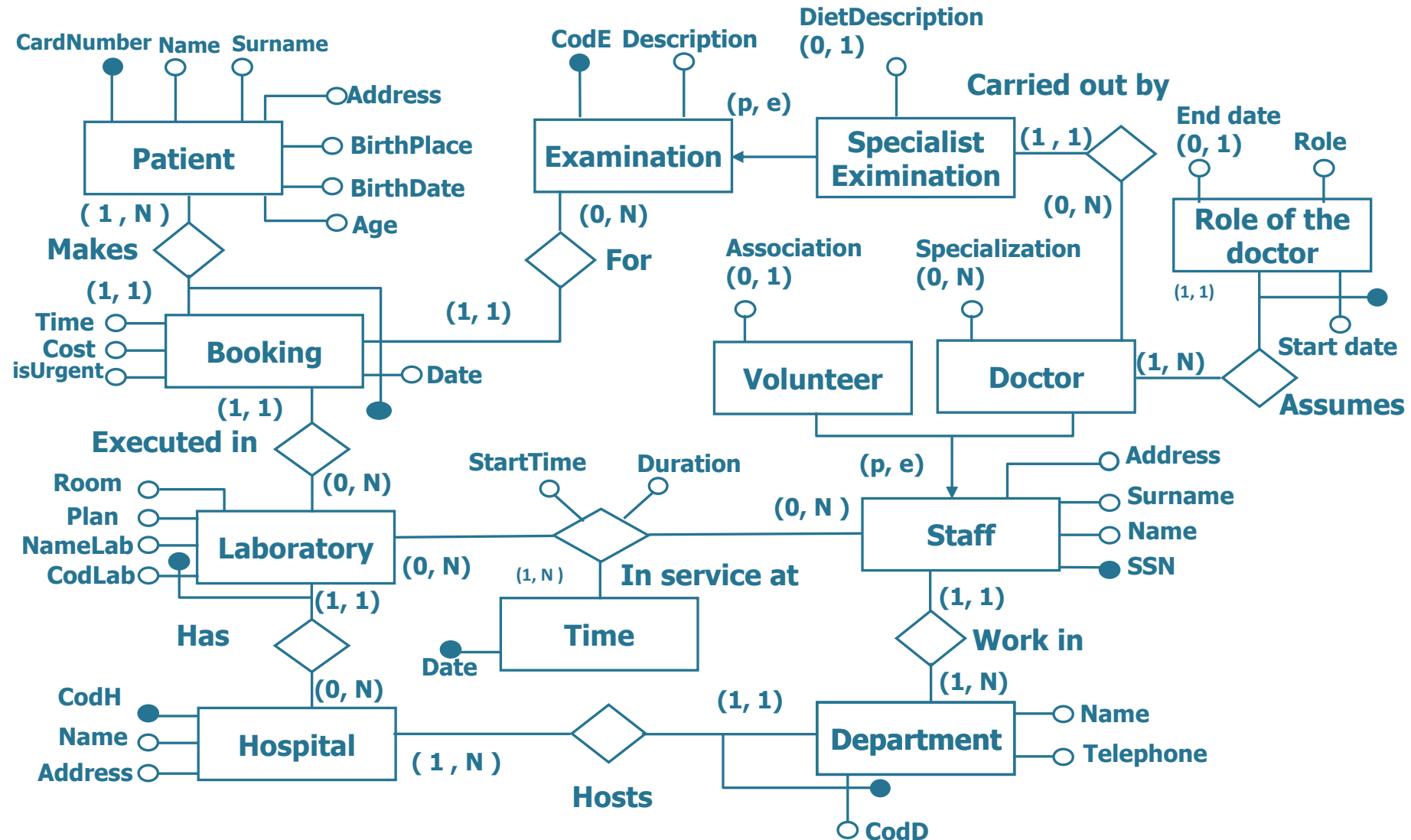
# Historicization of the role of the doctor



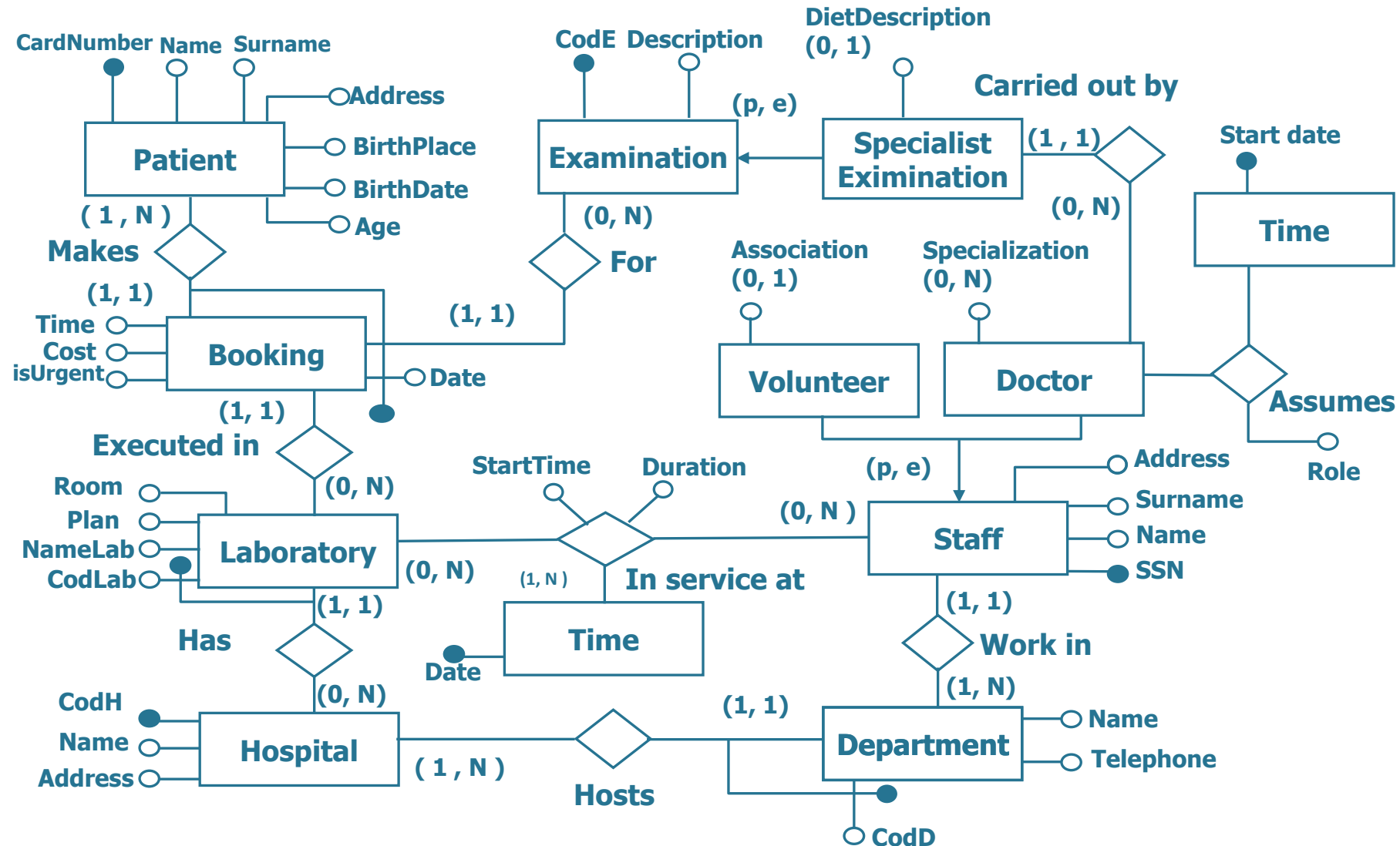
# Historicization of the role of the doctor



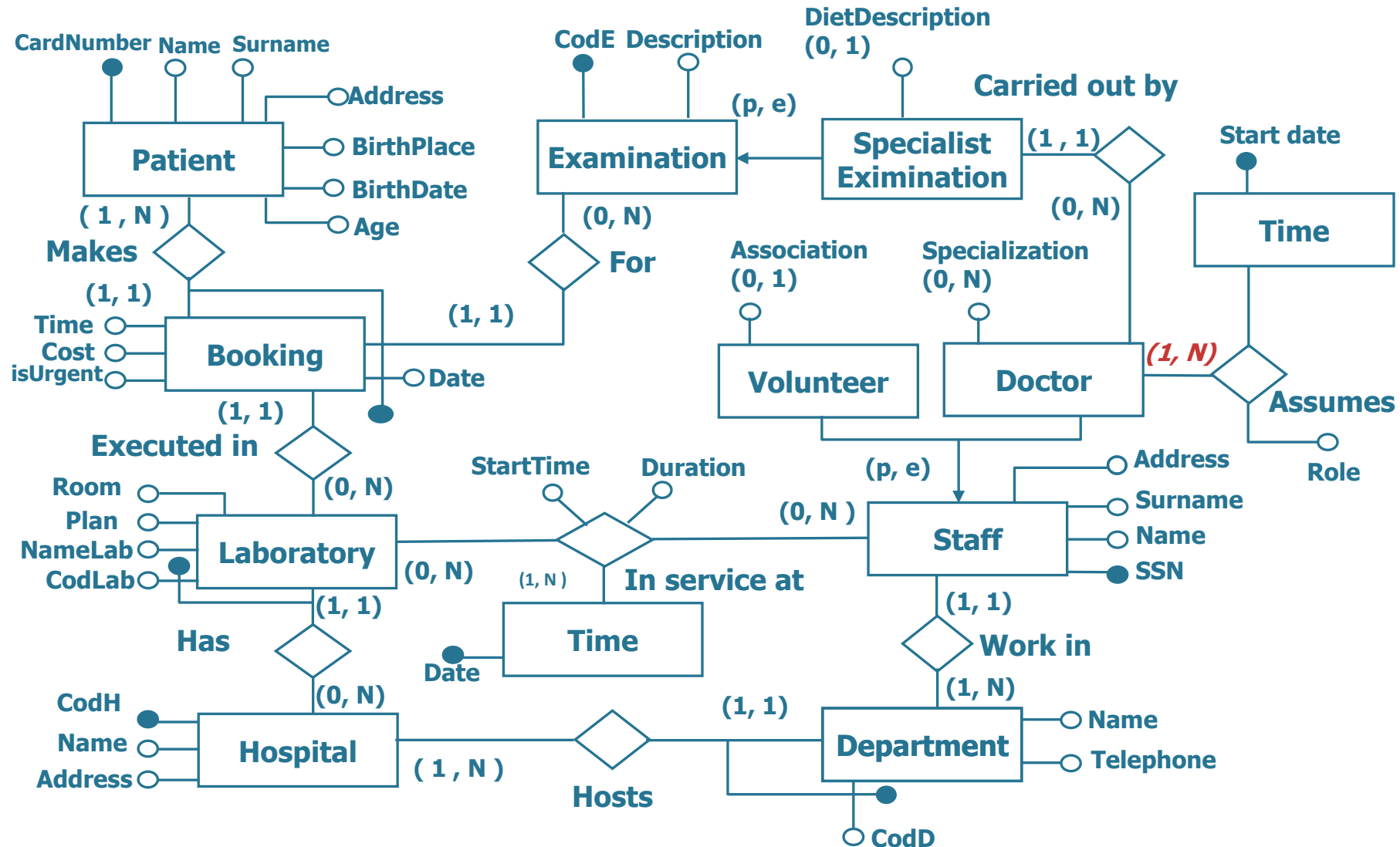
# Historicization of the role of the doctor



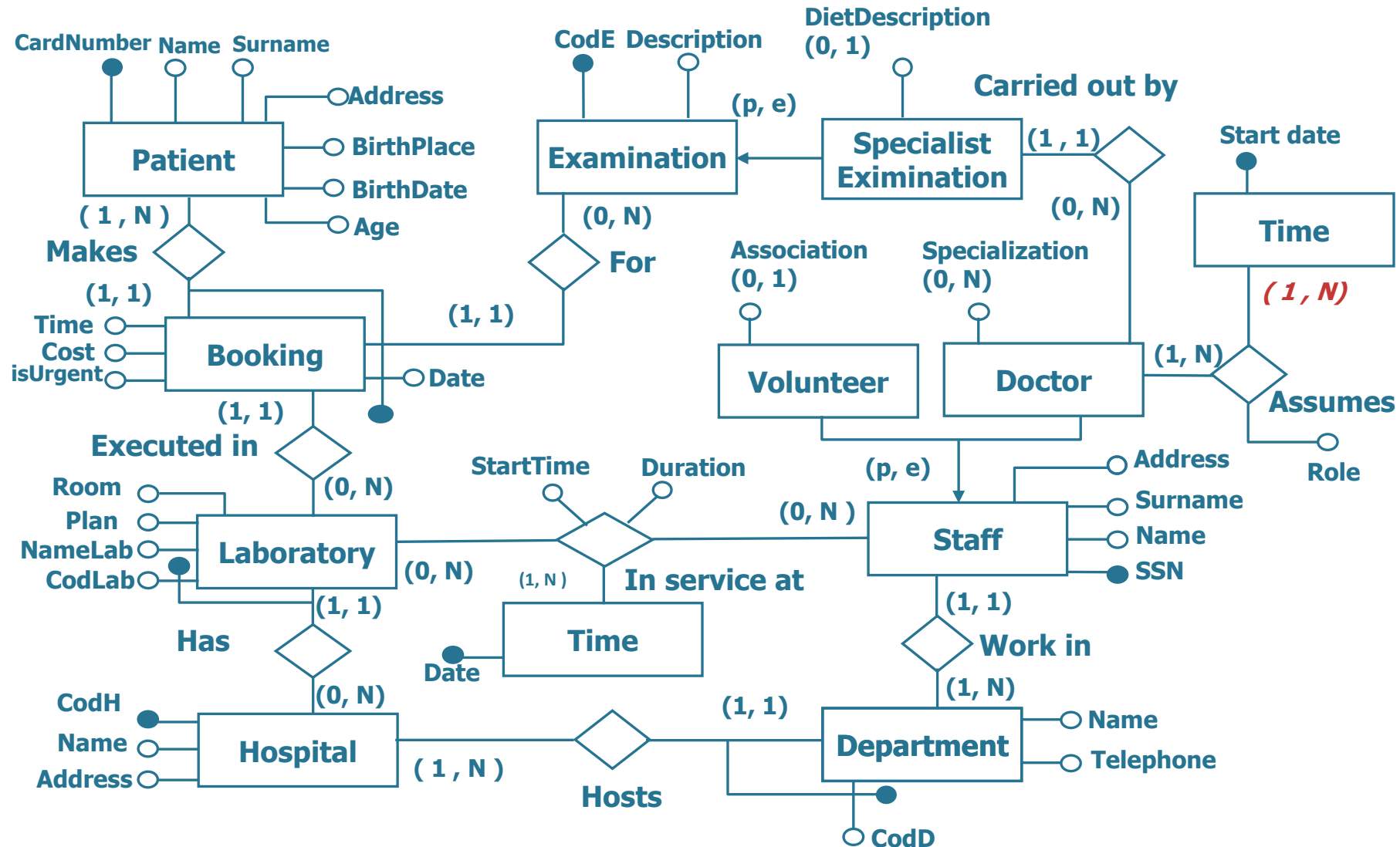
# Role history: alternative modeling



# Cardinality of the *Assumes* relationship



# Cardinality of the *Assumes* relationship



# Cardinality of the *Assumes* relationship

