Database design

Restructuring ER schema

Logical Design

- Introduction
- Removing composite attributes
- Removing multivalued attributes
- Removing generalizations
- Selection of primary identifiers
- Restructuring of the Entity-Relationship schema
- Partitioning of concepts

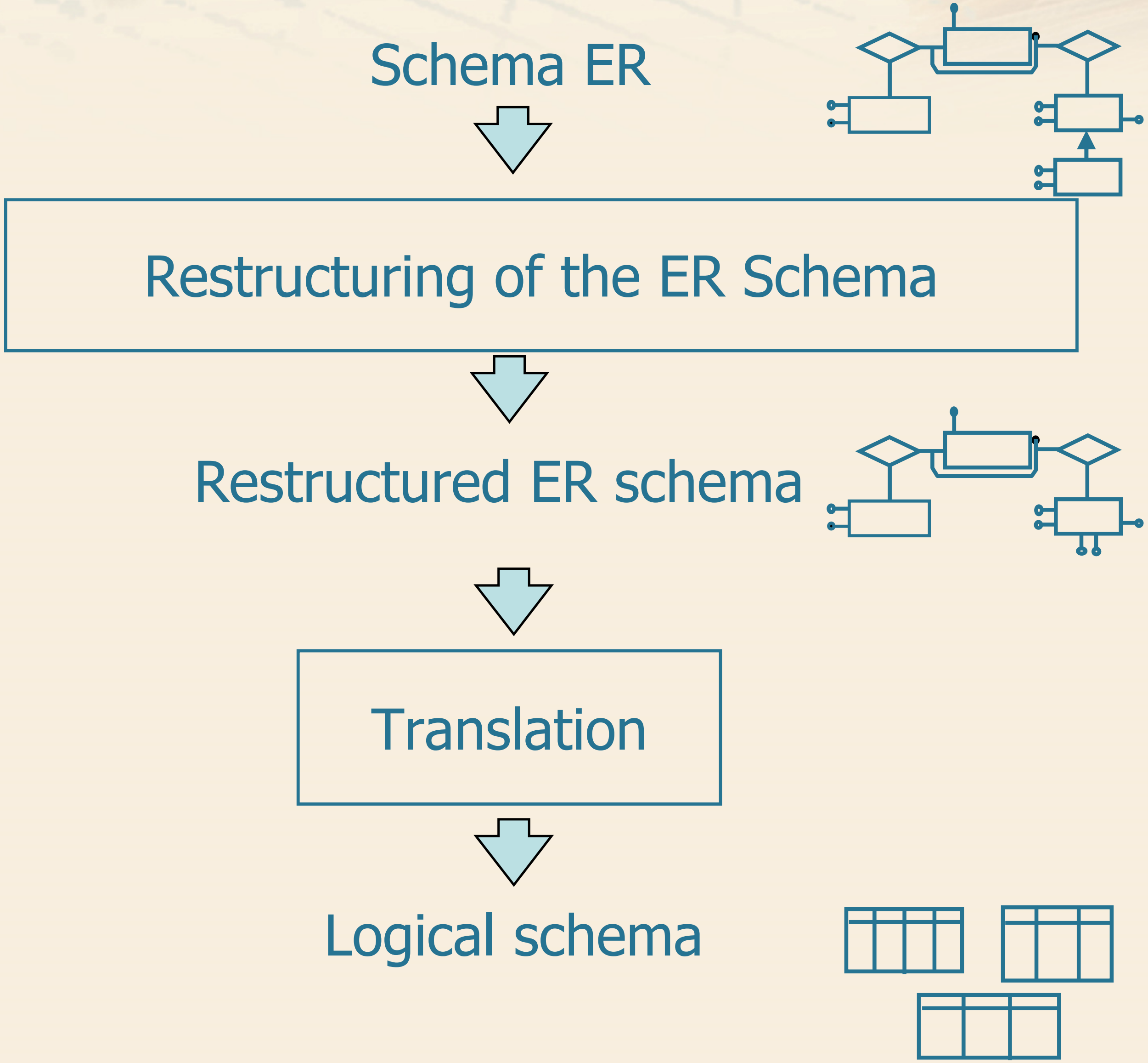
Logical Design

Introduction

Logical Design

- Select a logical model
  - in our case, the relational model.
- Main Goal
  - to construct a relational schema that correctly and efficiently represents all of the information described by an ER schema
- Not just a simple traslation
  - simplification of the scheme to make rappresentabile through the relational model.
  - optimization to increase the efficiency of queries.

Logical design steps



Schema ER

Restructuring of the ER Schema

Restructured ER schema

Translation

Logical schema

Logical Design

Removing composite attributes

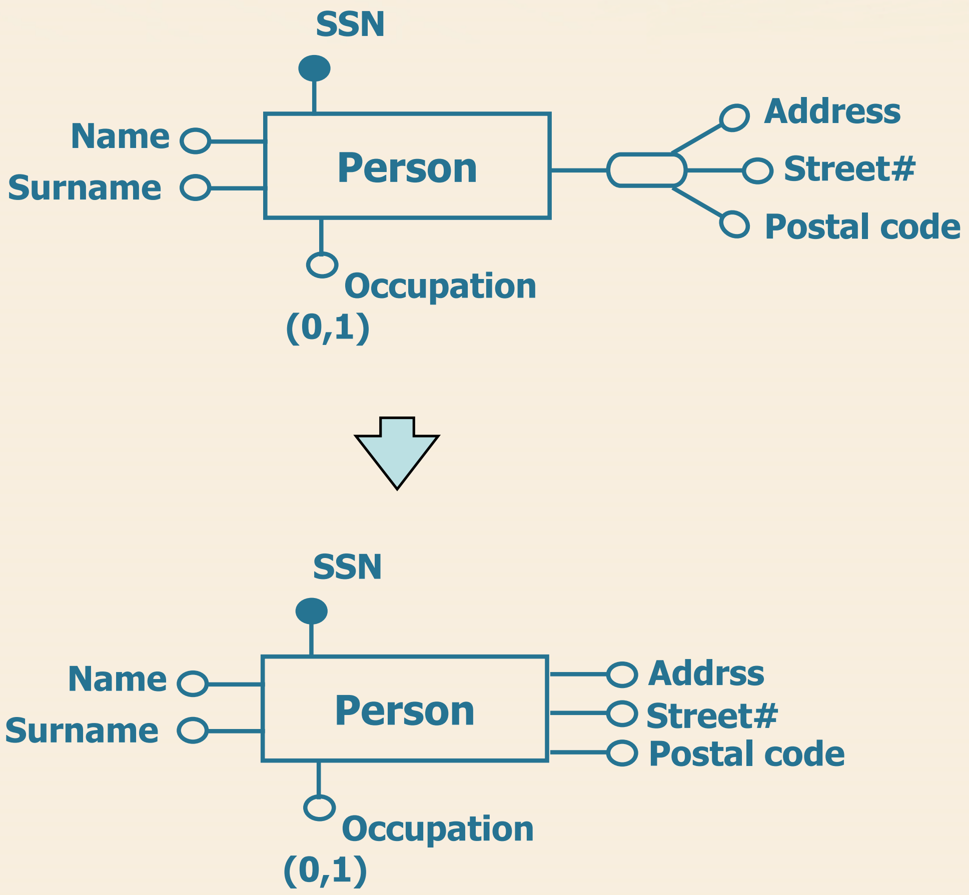
Removing composite attributes

- Composite attributes are not representable in the relational model.
- Two ways:

Removing composite attributes

- Composite attributes are not representable in the relational model.
- Two ways:
  - Split them in «individual» attributes.
    - useful if you need to access each attribute separately

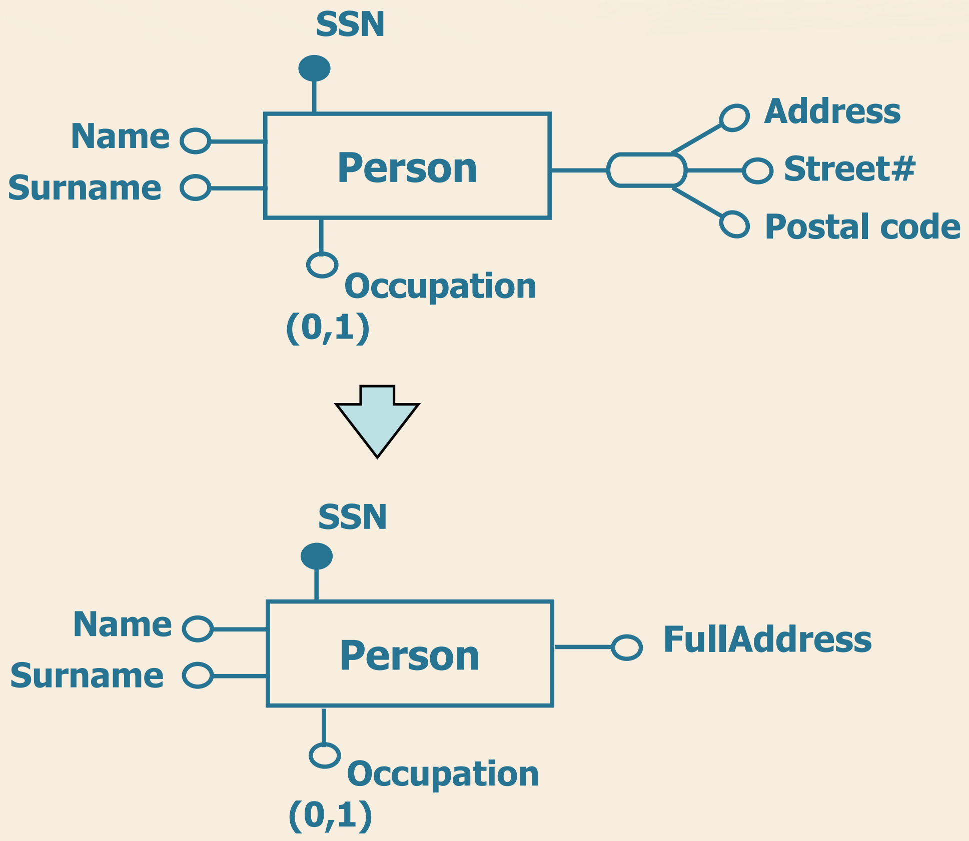
Split composite attributes



Removing composite attributes

- Composite attributes are not representable in the relational model.
- Two ways:
  - Split them in «individual» attributes.
    - useful if you need to access each attribute separately.
  - Use one attribute as a kind of «link».
    - useful if access to comprehensive information is enough

Example


SSN
Name
Surname
Person
Address
Street#
Postal code
Occupation
(0,1)
SSN
Name
Surname
Person
FullAddress
Occupation
(0,1)

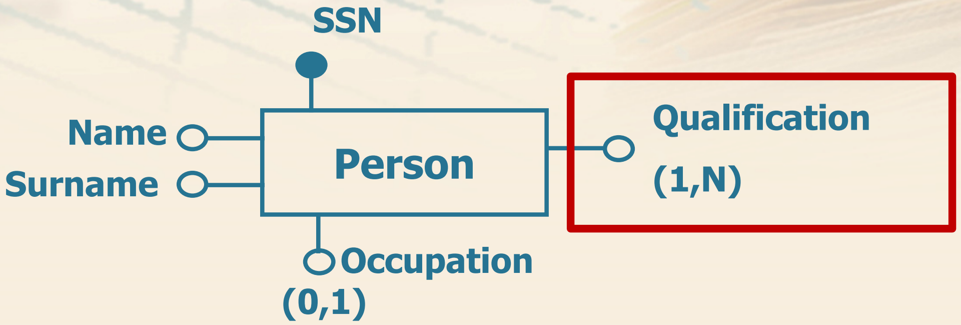
Logical Design

Removing multivalued attributes

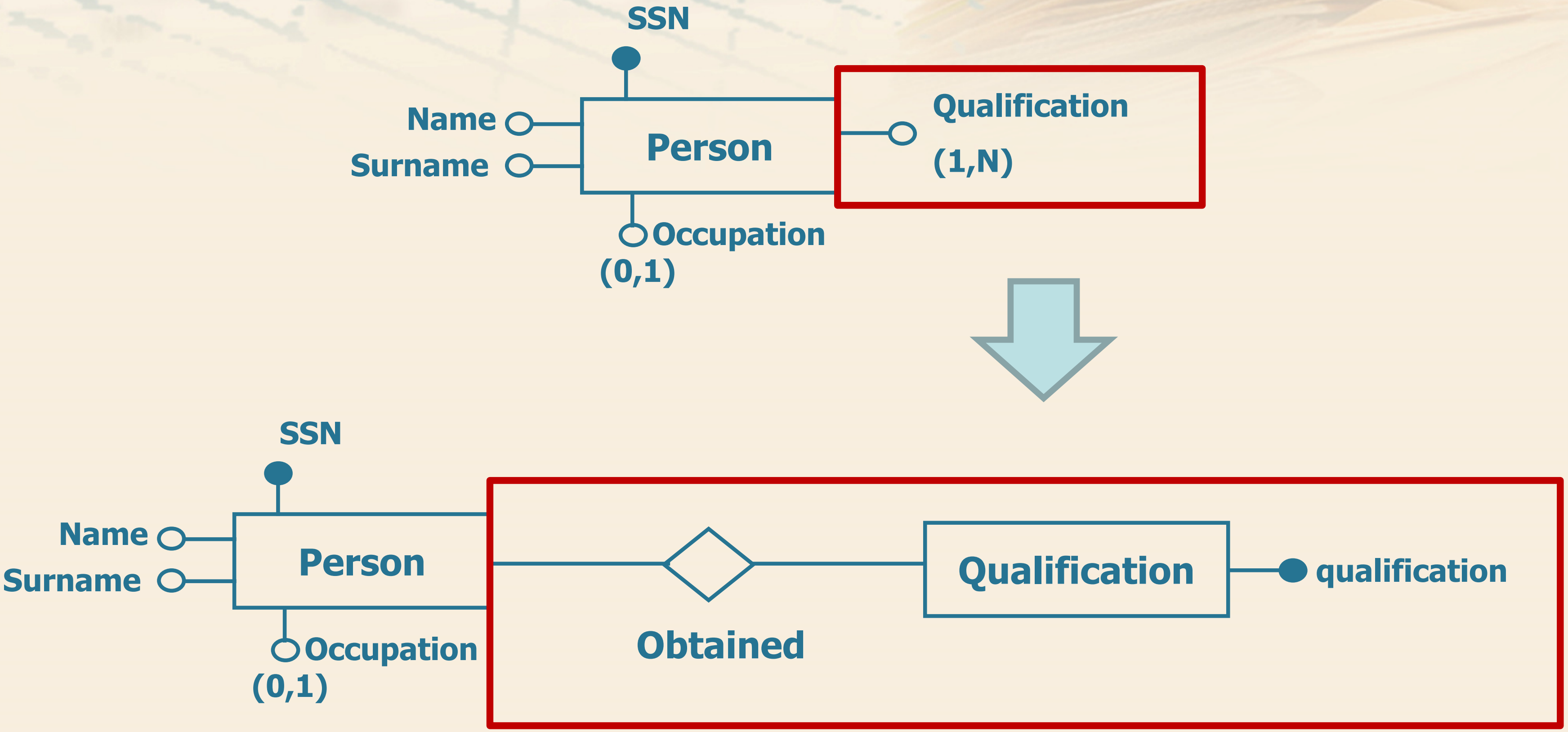
Removing multivalued attributes

- Multivalued attributes cannot be represented in the relational model.
- Mutivalued attribute is represented using a relationship between:
  - The original entity
  - A new entity
- Pay attention to the cardinality of the new relationship.

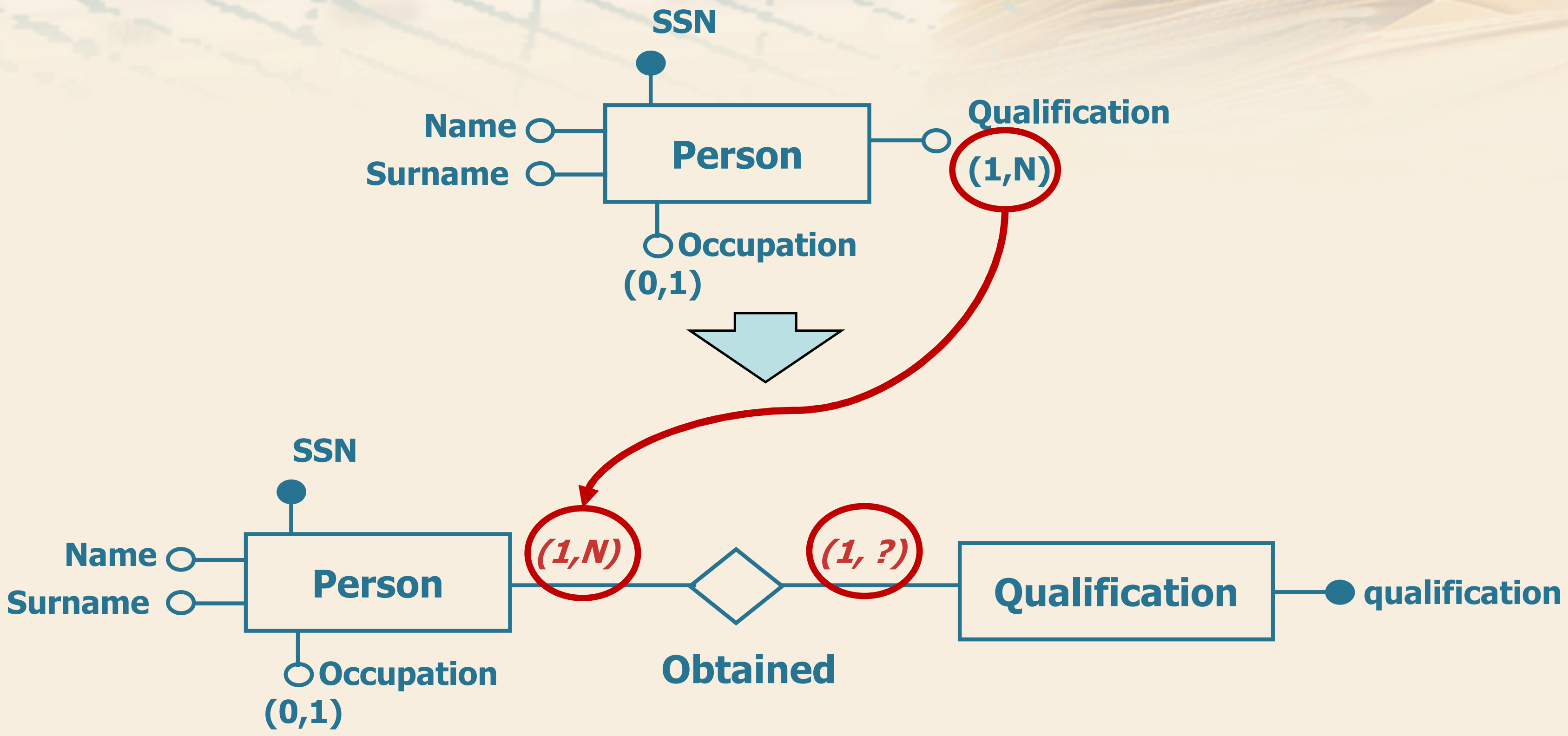
Removing multivalued attribute



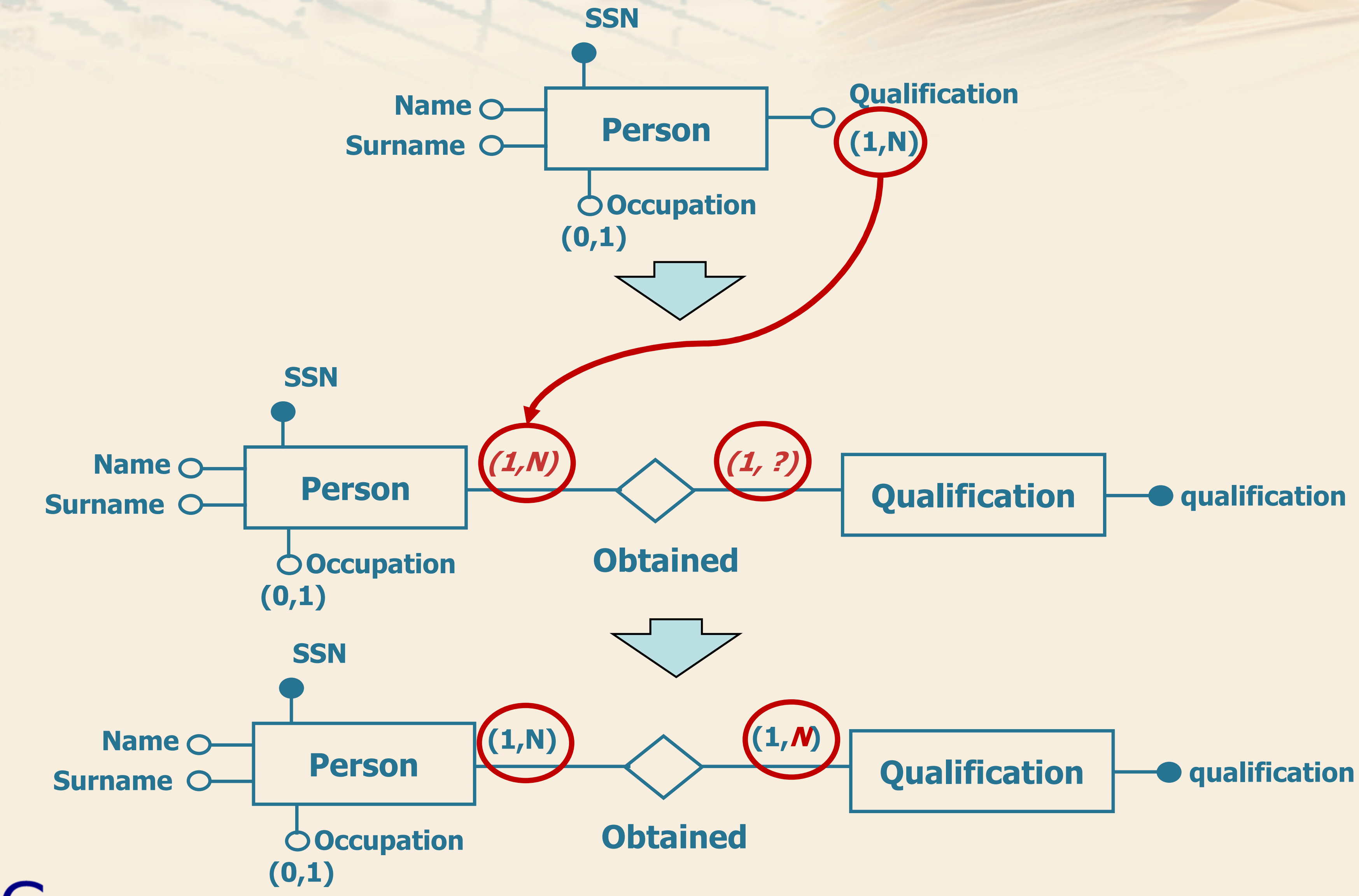
Cardinality of «Obtained»



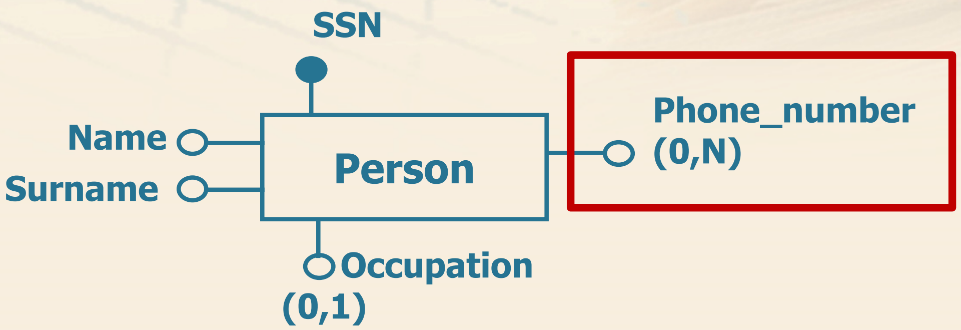
Cardinality of «Obtained»



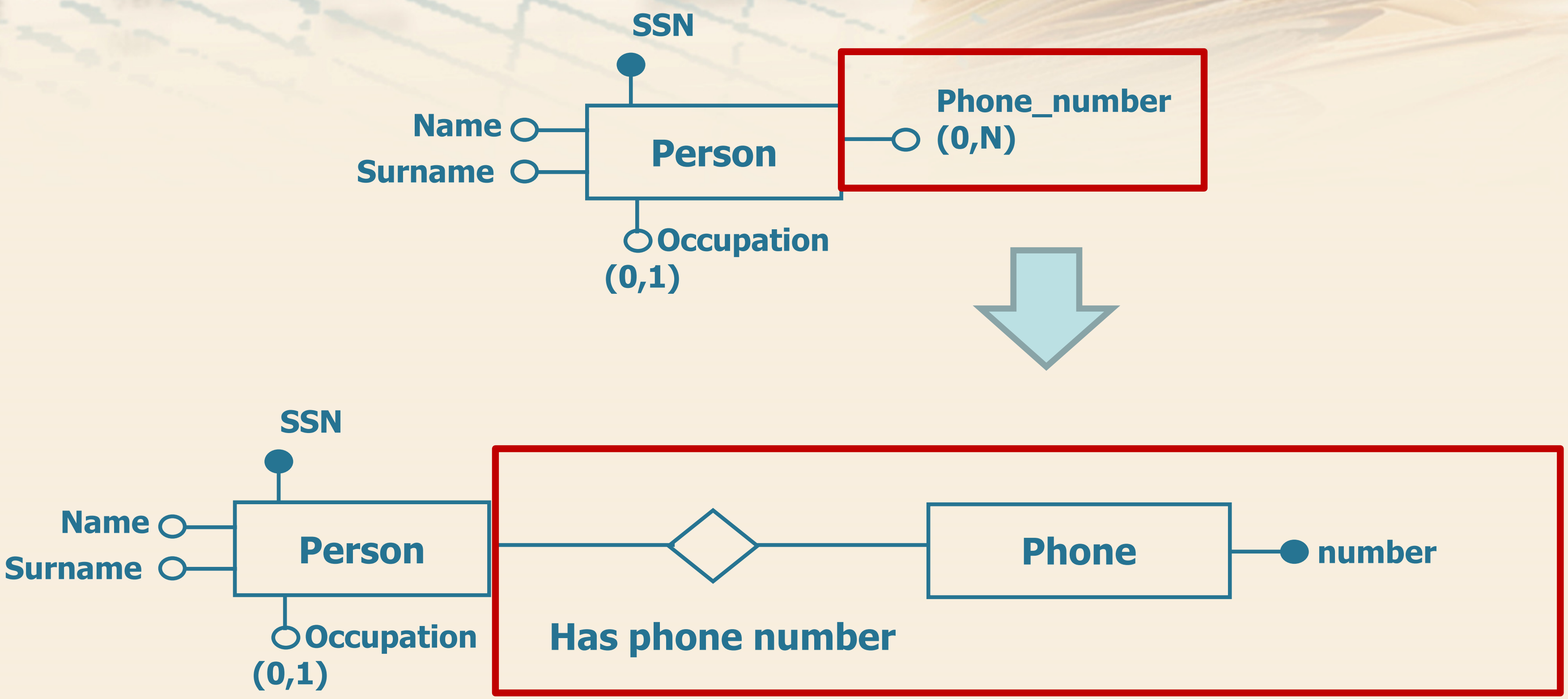
Cardinality of «Obtained»



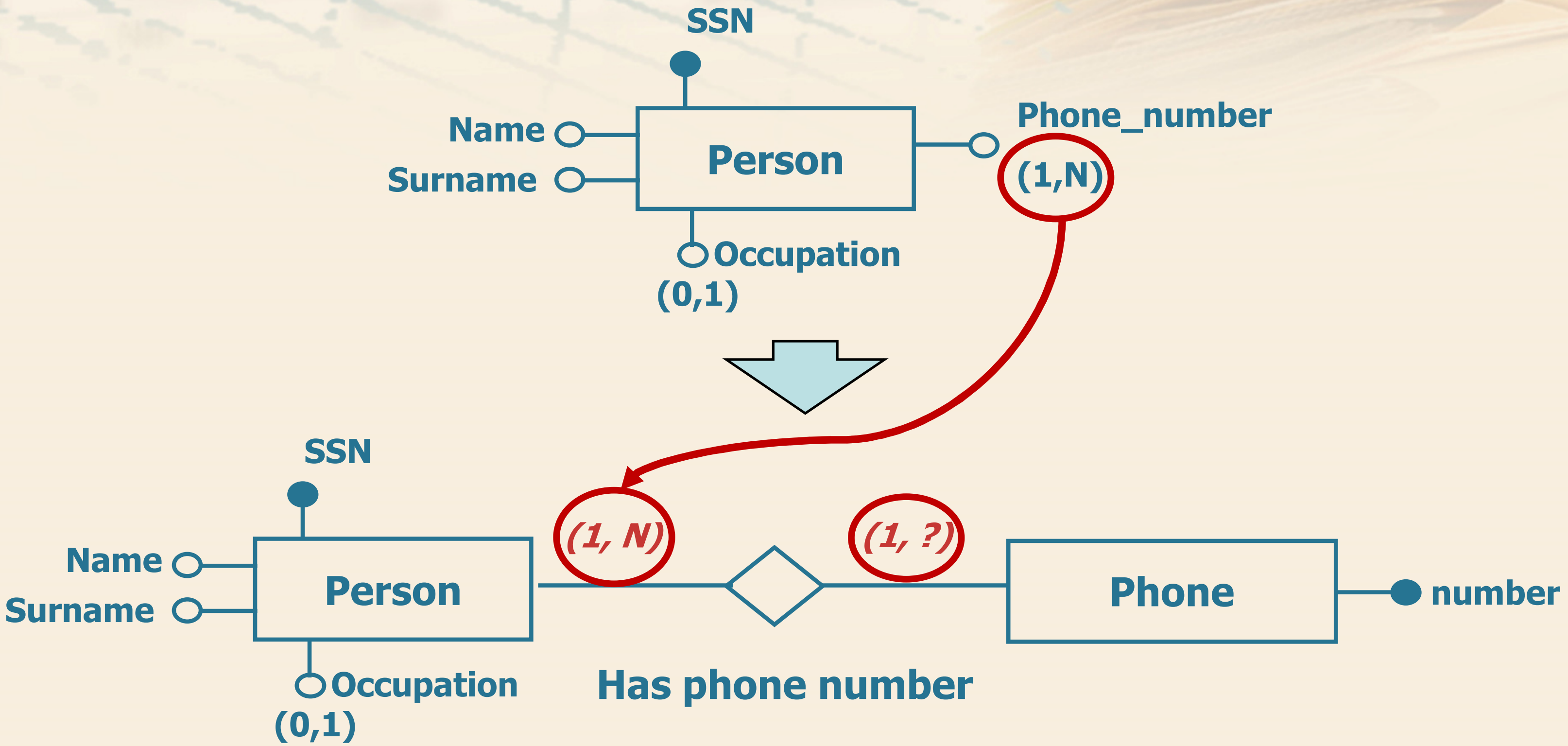
Removing multivalued attributes



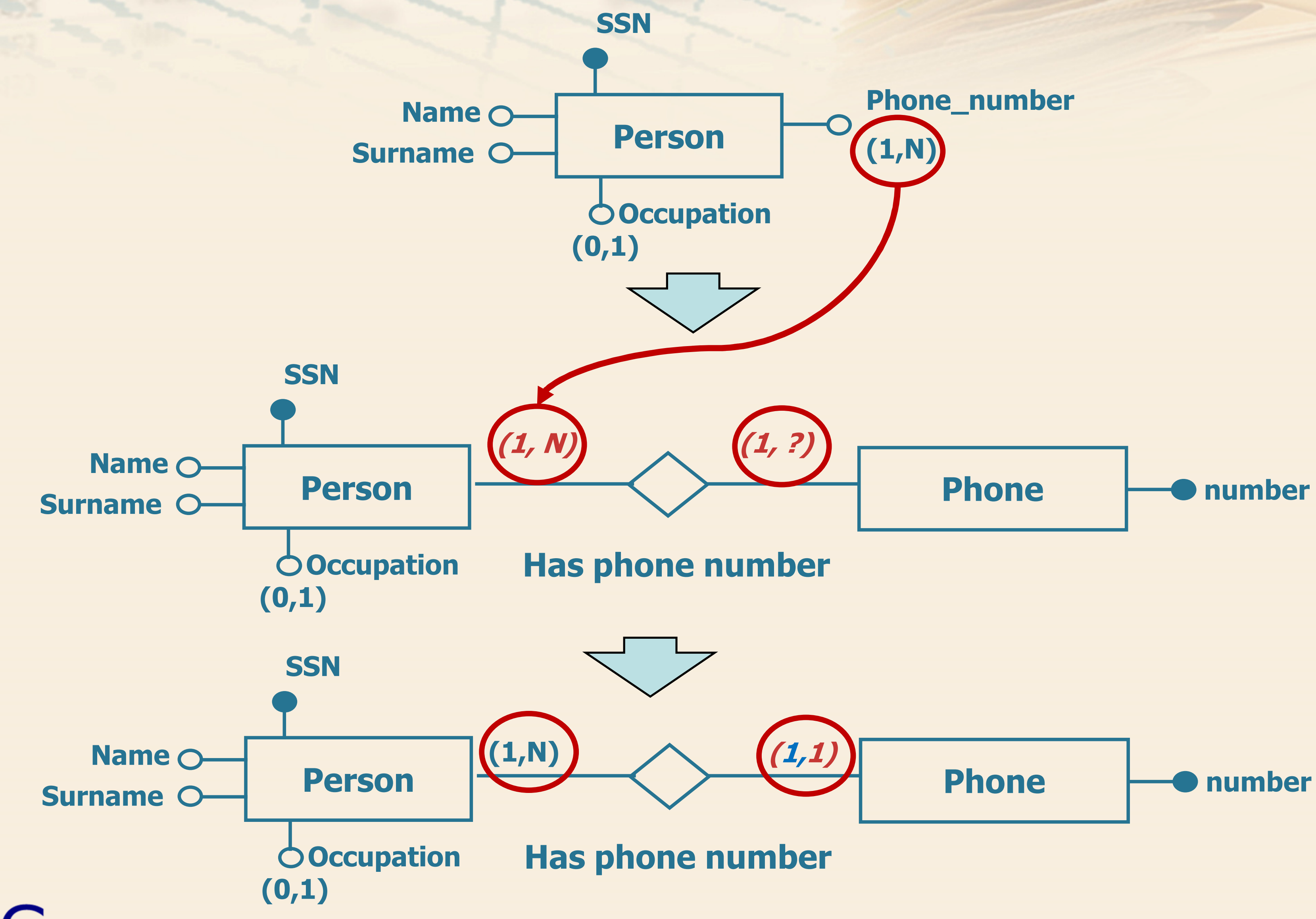
Removing multivalued attributes



Removing multivalued attributes



Removing multivalued attributes



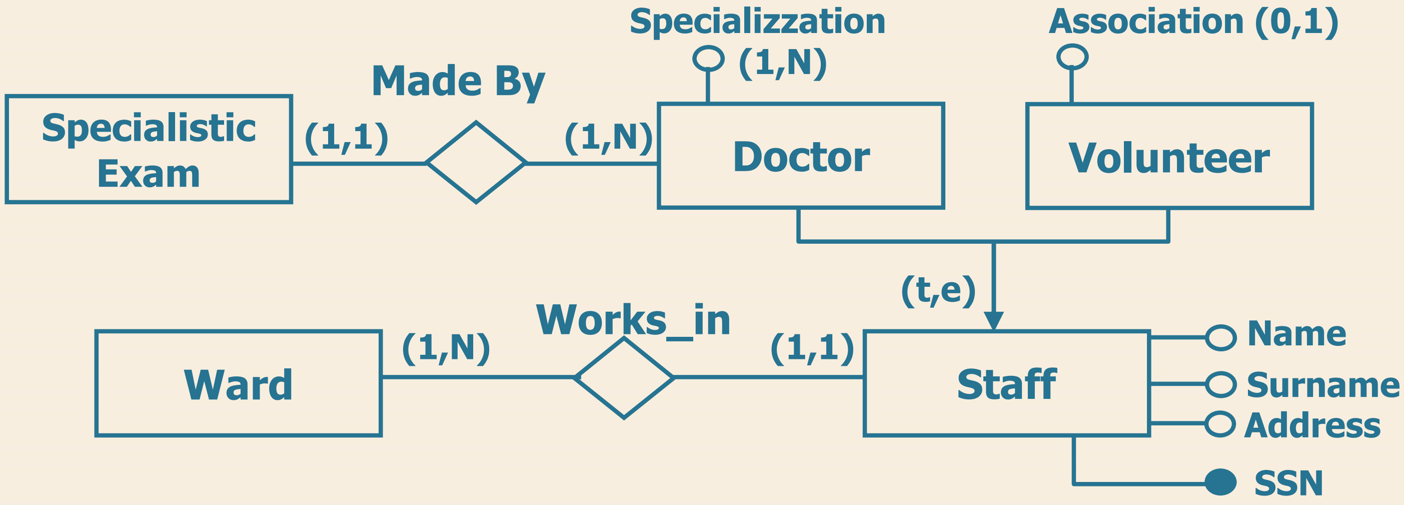
Logical Design

Removing Generalization

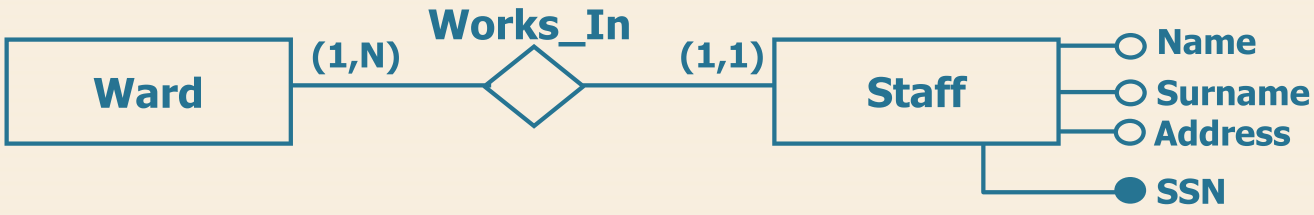
Removing Generalization

- The relational model does not allow the direct representation of generalizations of the ER model.
- We need, therefore, to trasform these into entities and relationships.
- Possible restructurings methods:
  - Child entities merged into parent entity
  - Parent entity merged into child entities
  - Generalization translated into relatioships

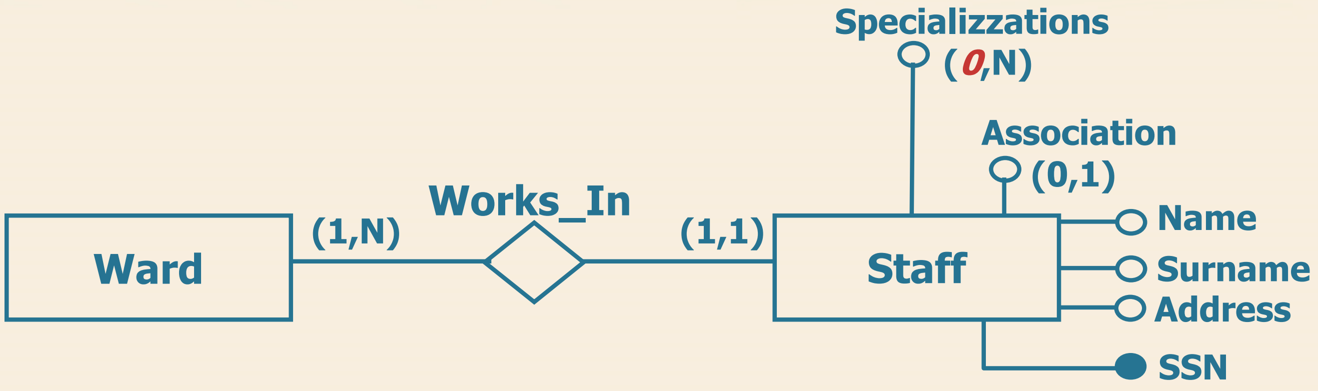
Example


Specializzation
(1,N)
Association (0,1)
Made By
Specialistic Exam
(1,1)
(1,N)
Doctor
Volunteer
(t,e)
Works_in
Ward
(1,N)
(1,1)
Staff
Name
Surname
Address
SSN

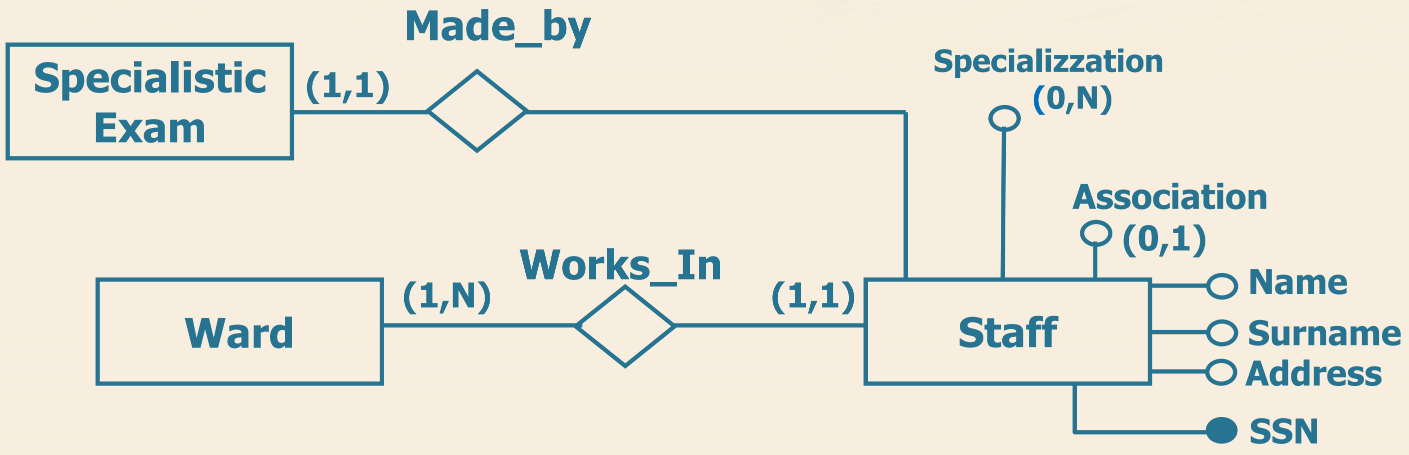
Child->Parent



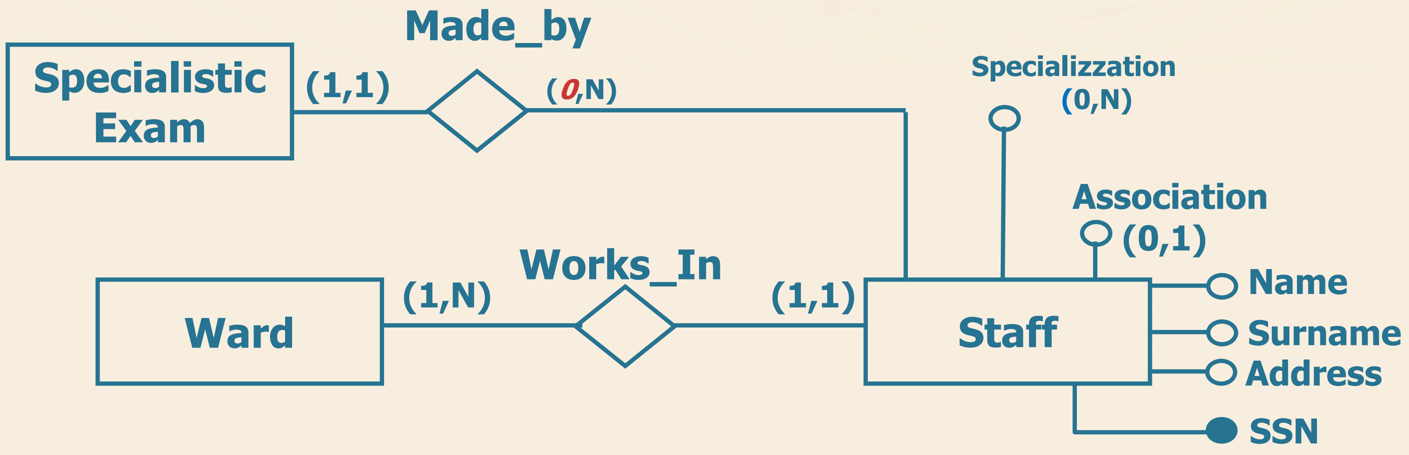
Child entities' attributes



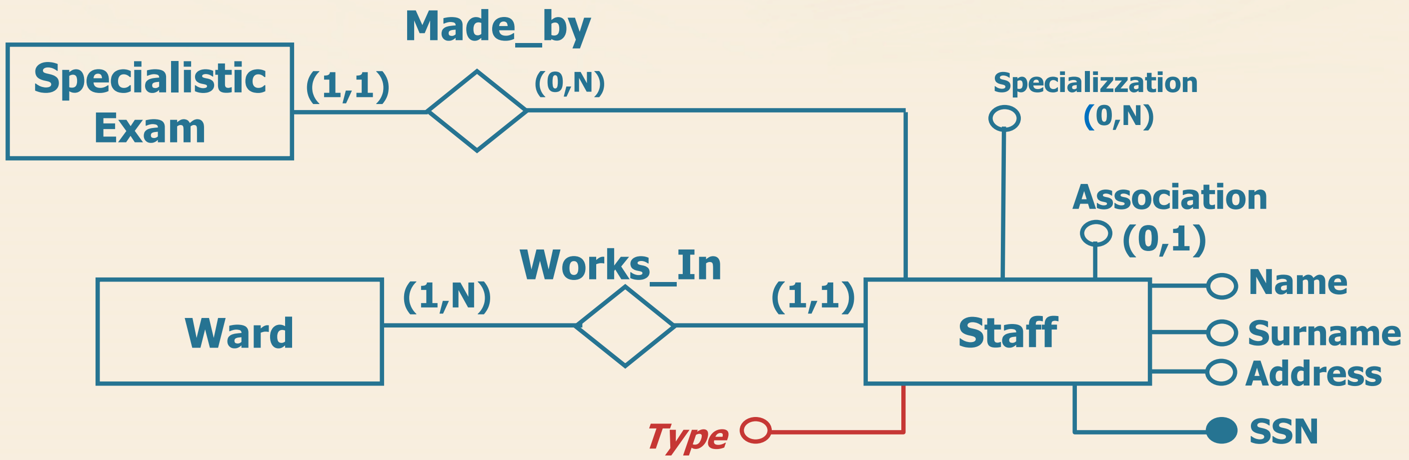
Relations with child entities



Relations with child entities

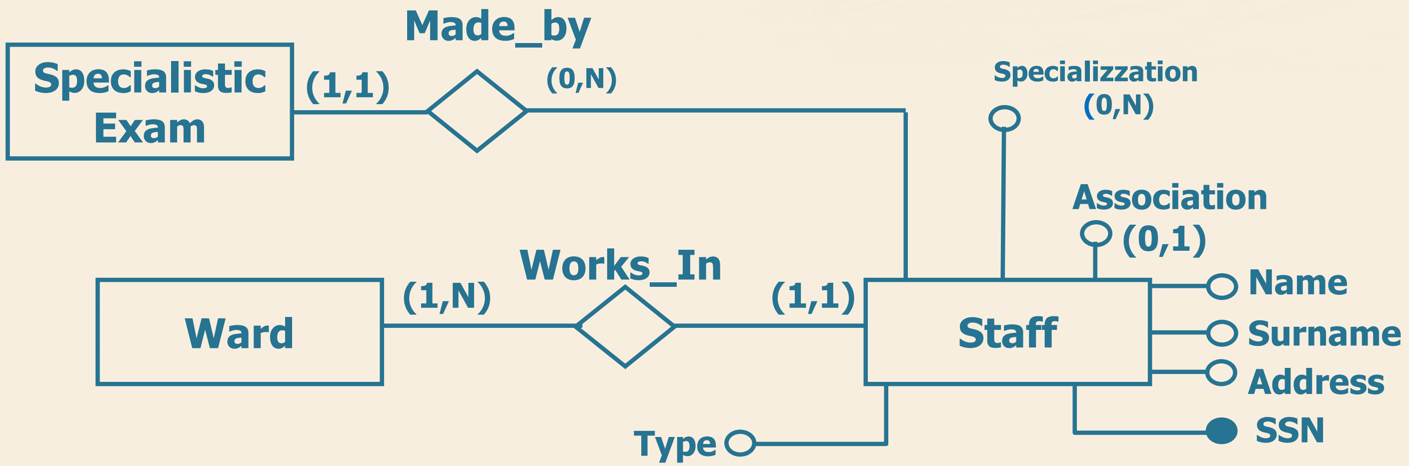


The «Type» attribute



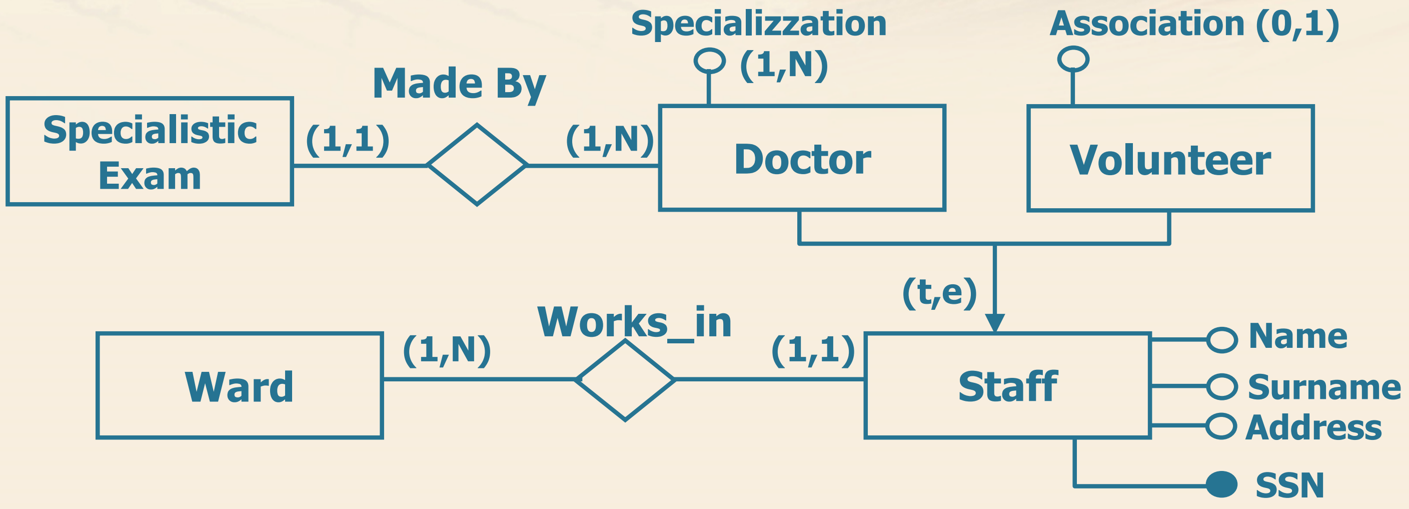
- «Type» allows to indicate the original entity

Child->Parent

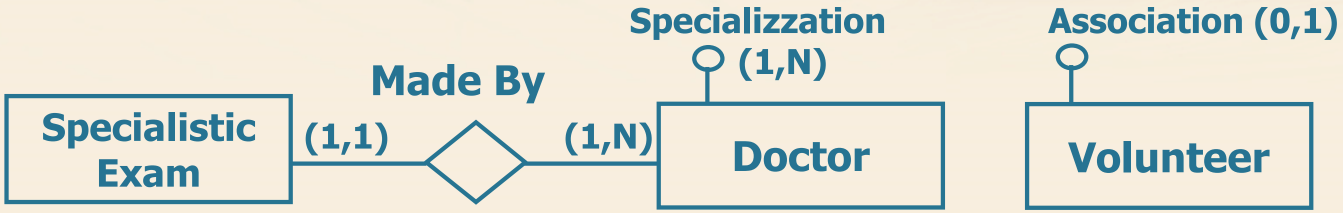


- Always usable
  - in case of overlapped entities, many combinations are possible as Type values

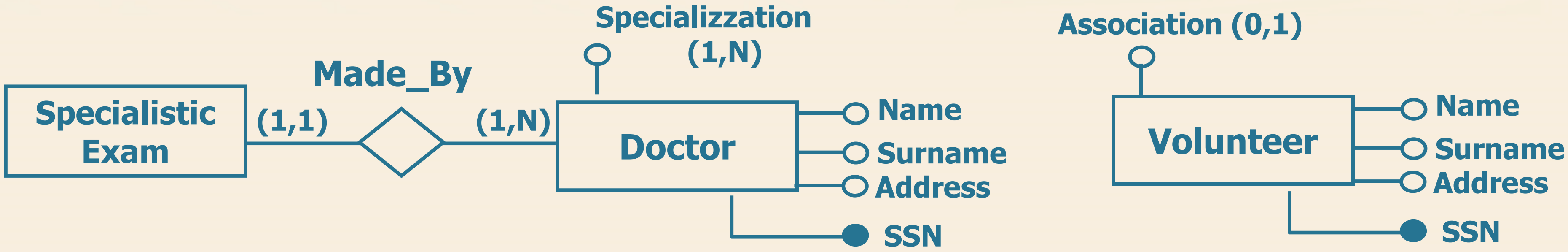
Example


Specializzation
(1,N)
Association (0,1)
Made By
Specialistic Exam
(1,1)
(1,N)
Doctor
Volunteer
(t,e)
Works_in
Ward
(1,N)
(1,1)
Staff
Name
Surname
Address
SSN

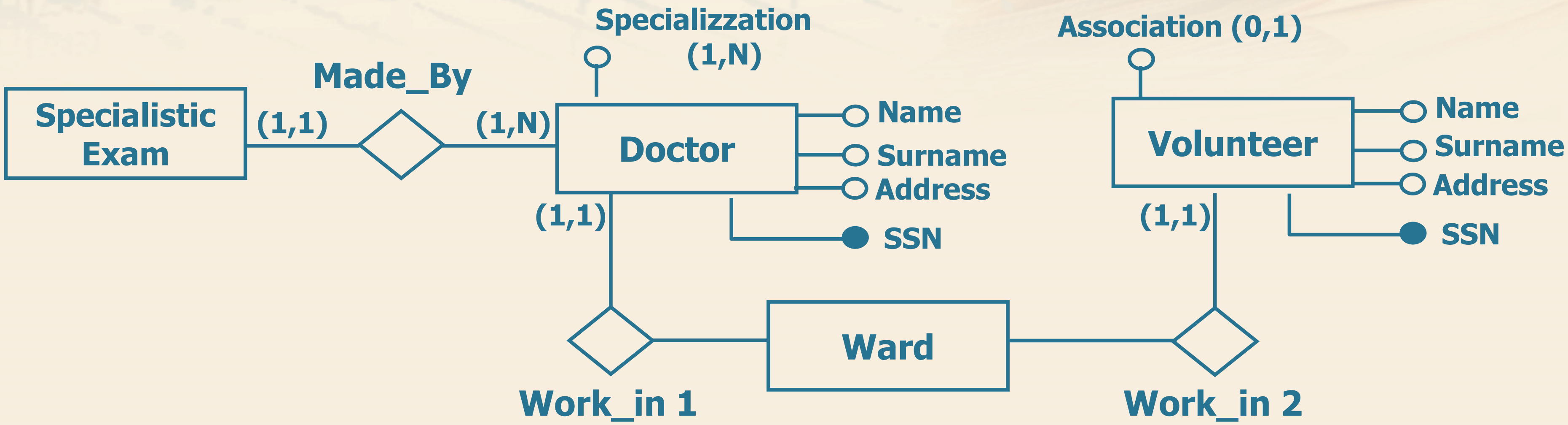
Parent->Child



Parent’s attributes

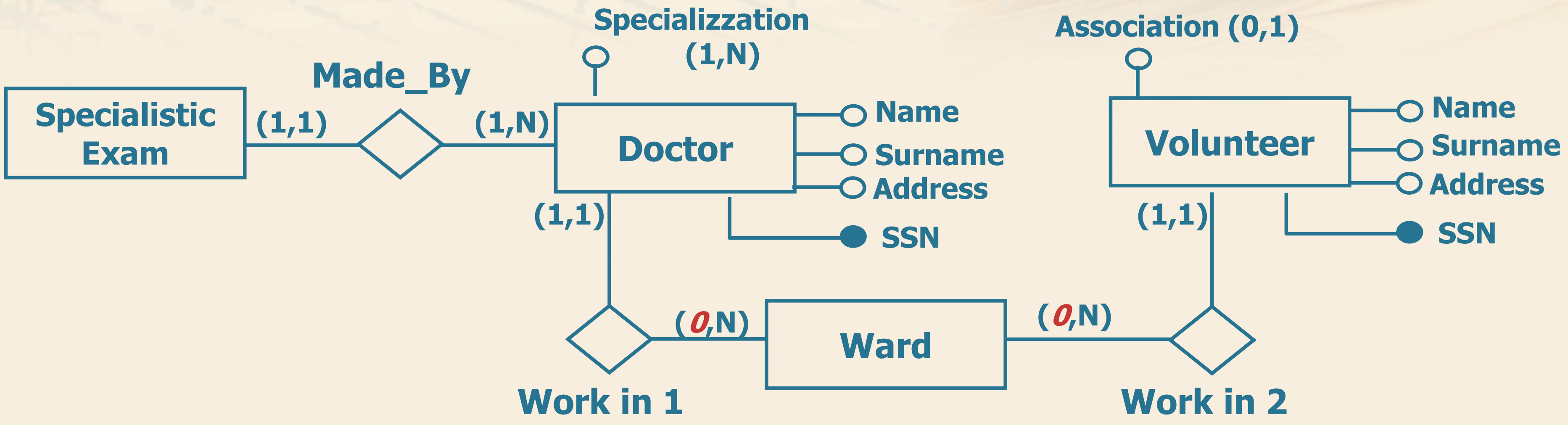


Relationships with parent



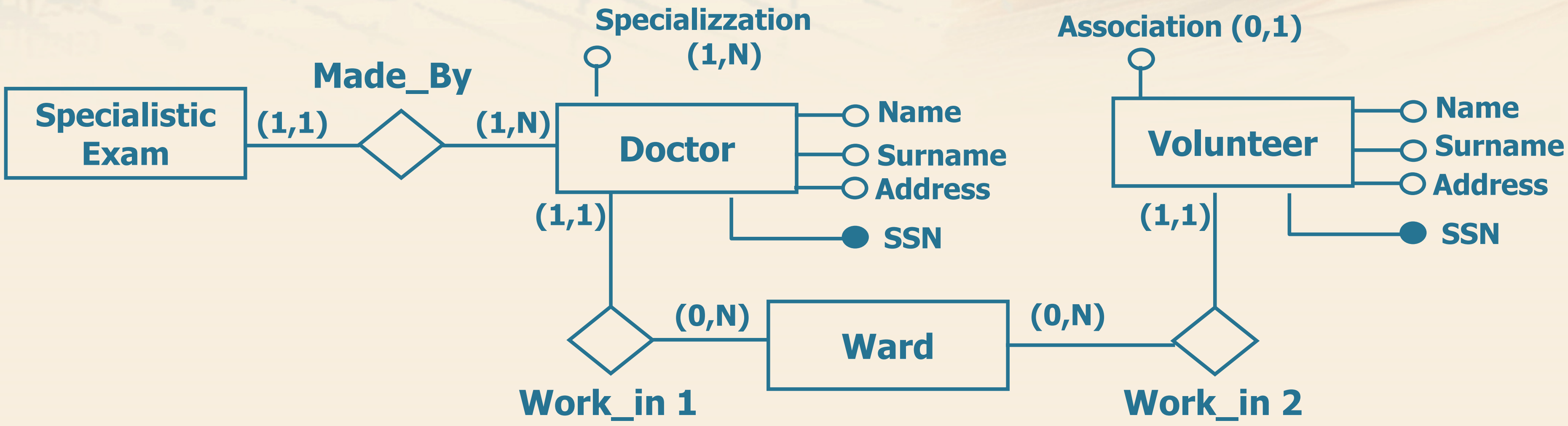
- Relations with the parent entity need to be split

Cardinality of «work in»



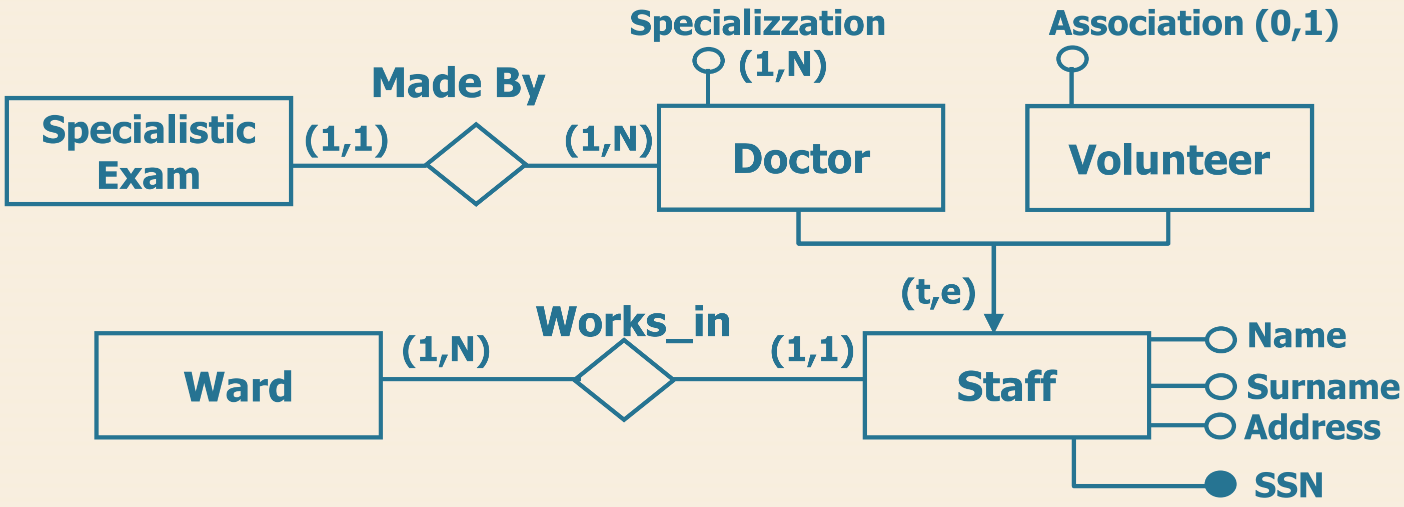
- Relations with the parent entity need to be split

Parent -> Child

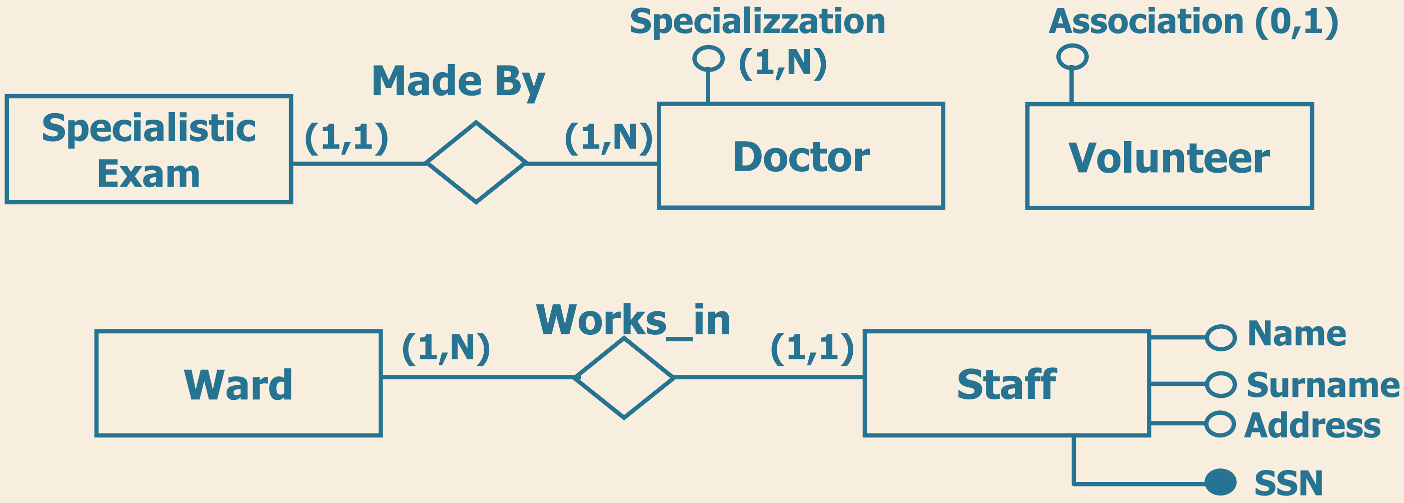


- It Cannot be used for **partial** generalizzation
  - However, we can trasform partial generalizzation into total, adding a new entity called «Others»
- It Cannot be used for **overlapping** generalizzation
  - Duplicate identifiers cause problems.

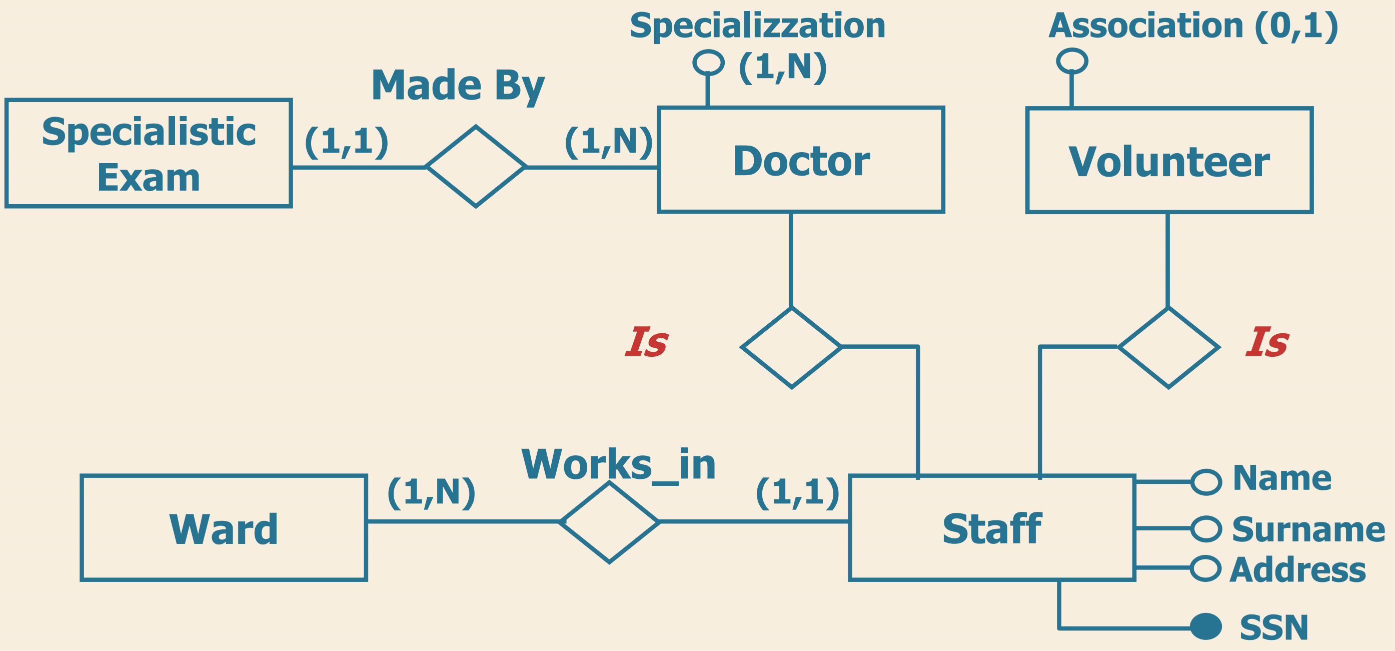
Example


Specialistic Exam
Made By
(1,1)
(1,N)
Specializzation
(1,N)
Association (0,1)
Doctor
Volunteer
(t,e)
Ward
(1,N)
Works_in
(1,1)
Staff
Name
Surname
Address
SSN

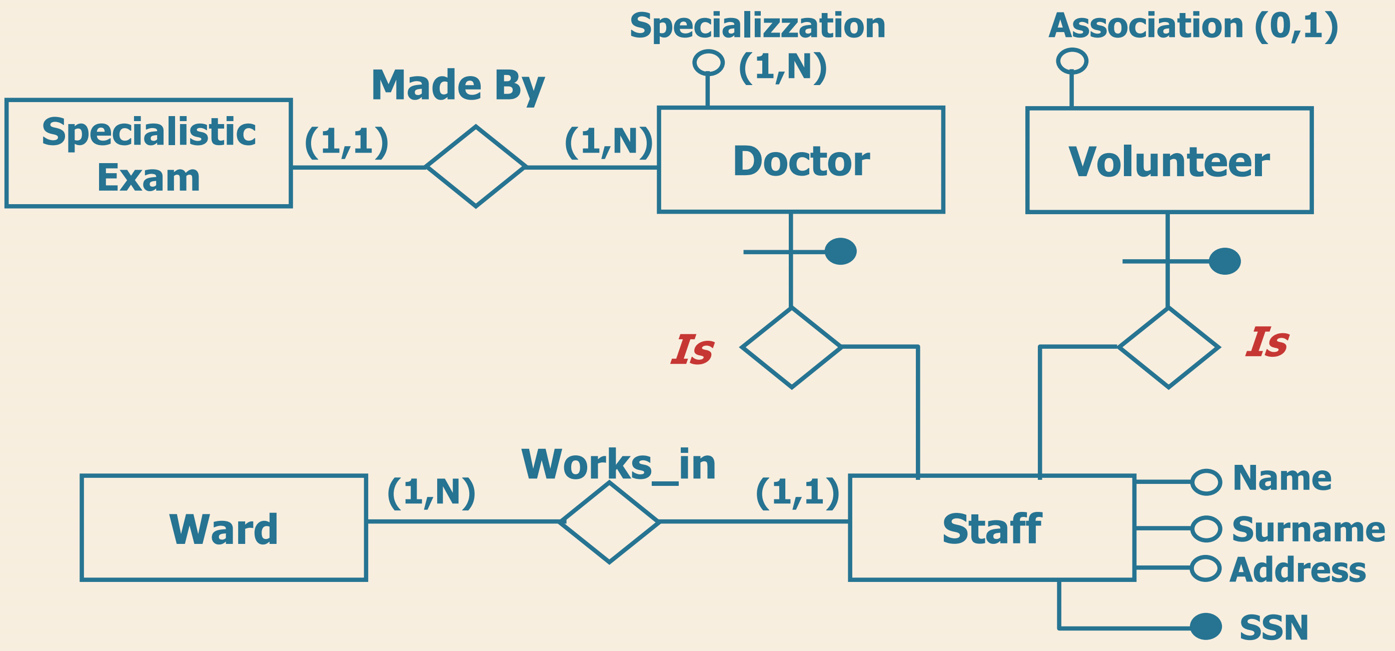
Relation between parent and child entities



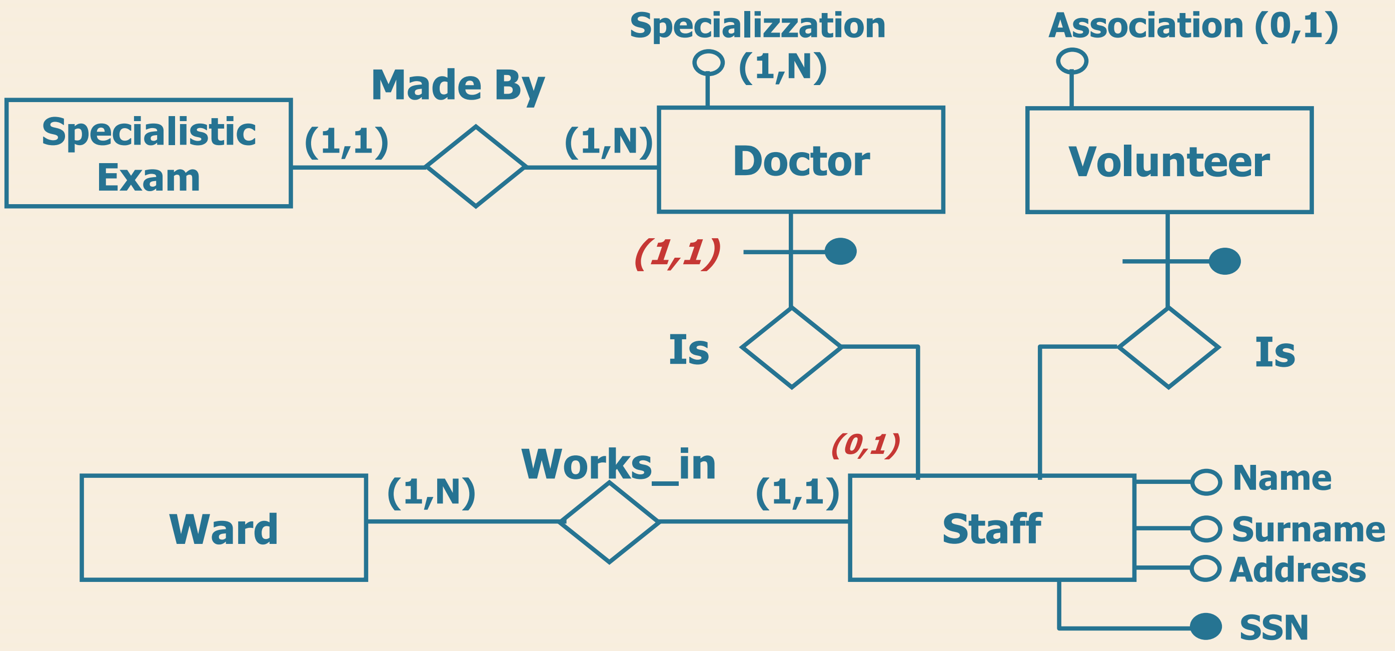
Relationship between parent and child entities



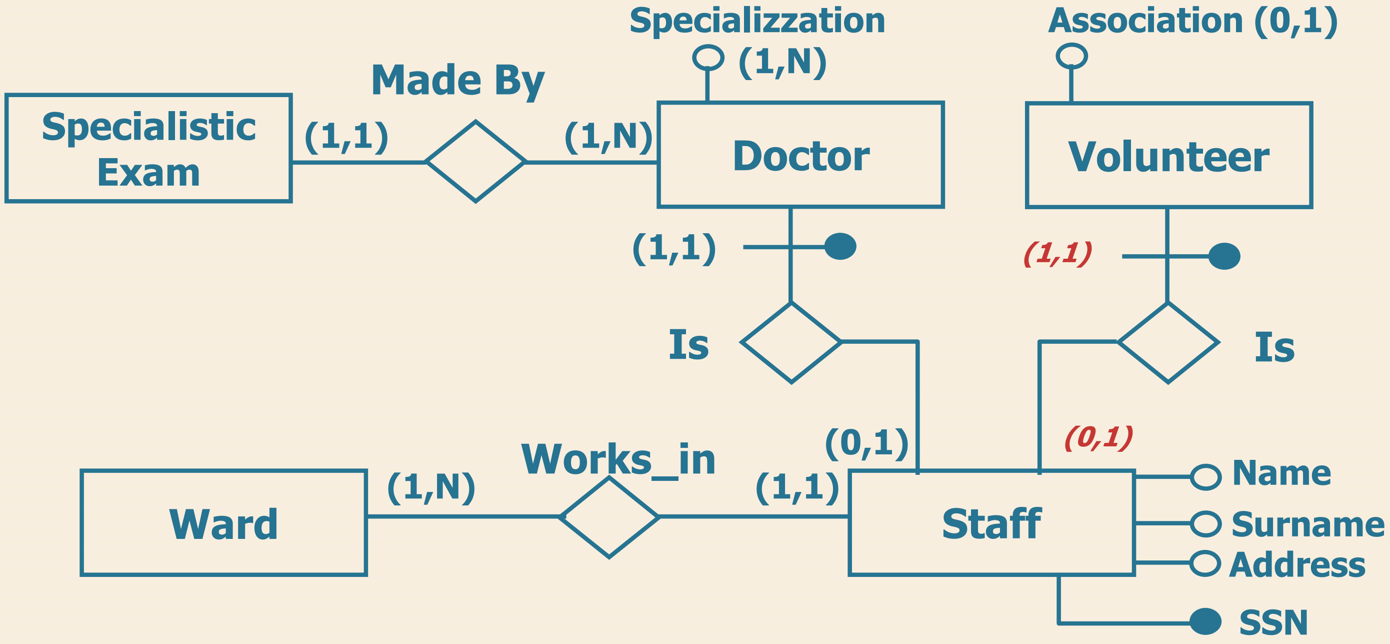
Child entities' identification



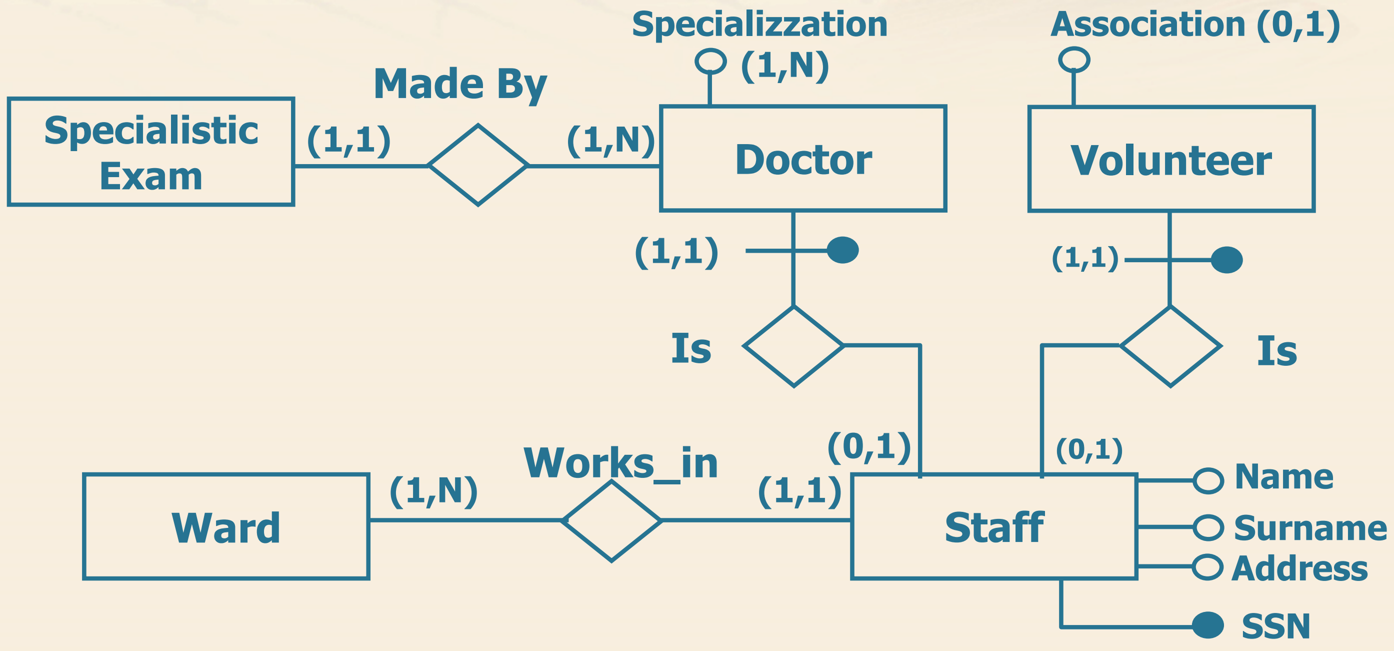
Cardinality of «is»



Cardinality of «is»



Generalization translated into relatioships



- This solution can always been adopted
  - but rebuilding the original information could be expensive

Assessment of alternatives

- Merging child entities into parent entity is useful when:
  - The operations involve the occurrences and the attributes of child and parent entitites more or less in the same way (optimize data access).

Assessment of alternatives

- Merging parent entity into child entities is useful when:
  - The generalizzation is «total»
  - there are operations that refer only to occurrences of child entities and so they make distinctions between these entities (optimize data access).

Assessment of alternatives

- The various options can be combined
  - there are operations that refer only to occurrences of some child entities (optimize data access).

Assessment of alternatives

- In presence of hieracy:
  - Procede in the same way
  - Start from the lower levels.

Logical Design

Selection of primary identifiers

Selection of primary identifiers

- It is necessary to define the *primary key*.
- The criteria for this decision are as follows:
  - Attributes with null values cannot be selected as primary identifiers.
  - Just **one** or few attributes
  - An internal identifier with few attributes is preferable to an external one
  - It is used by many operations to access the occurrences
- Is possible to introduce a further attribute to the entity, often called *codes*

Restructuring of an ER schema

ER scheme restructuring

- Implementation aspects
  - This is not a conceptual schema
- Goals
  - removing of costructs for which there is no direct representation in the relational model.
  - Optimize data access.

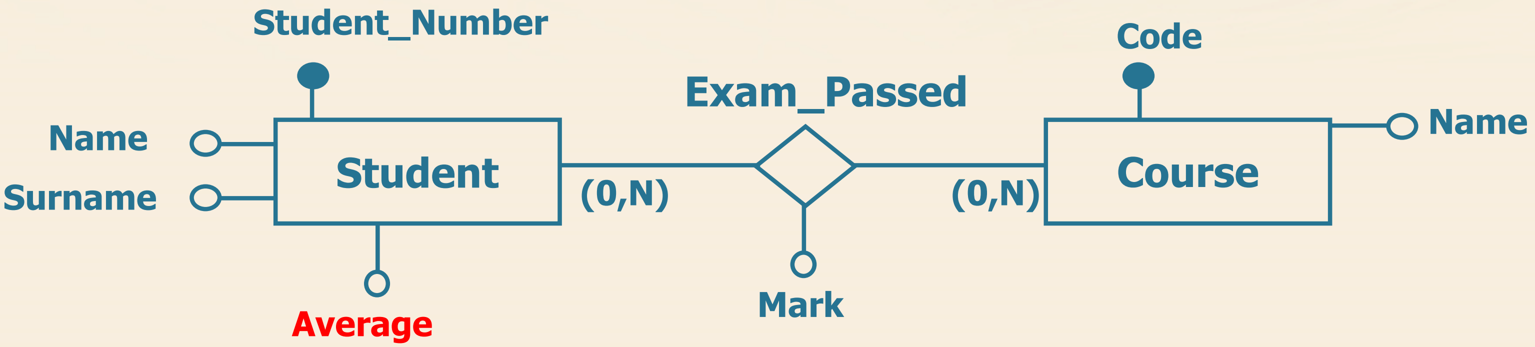
Restructuring tasks

- Analysis of redundancies
- Removing generalizzations
- Partitioning and merging of entities and relations
- Selection of primary identifiers

Analysis of redundancies

- Topic
  - To represent informations that can be derived from other data. Decide whether to keep or remove them.
- An advantage
  - Speed up and simplify queries
- Disadvantages
  - increased complexity of updates
  - slowing down of updates
  - more storage space required.

Redundant attribute: example



- In this schema the attribute Average is redundant
  - It is useful for speeding up queries to calculate studendt’s average.
  - if kept, the redundancy indication must be added in the relational schema.

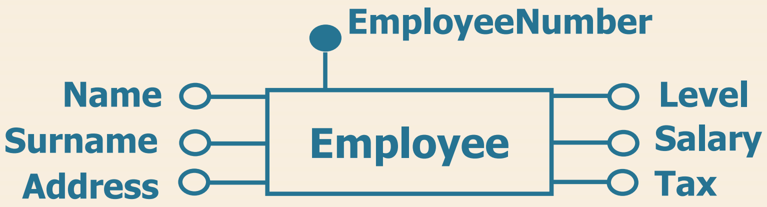
Logical Design

Partitioning of concepts

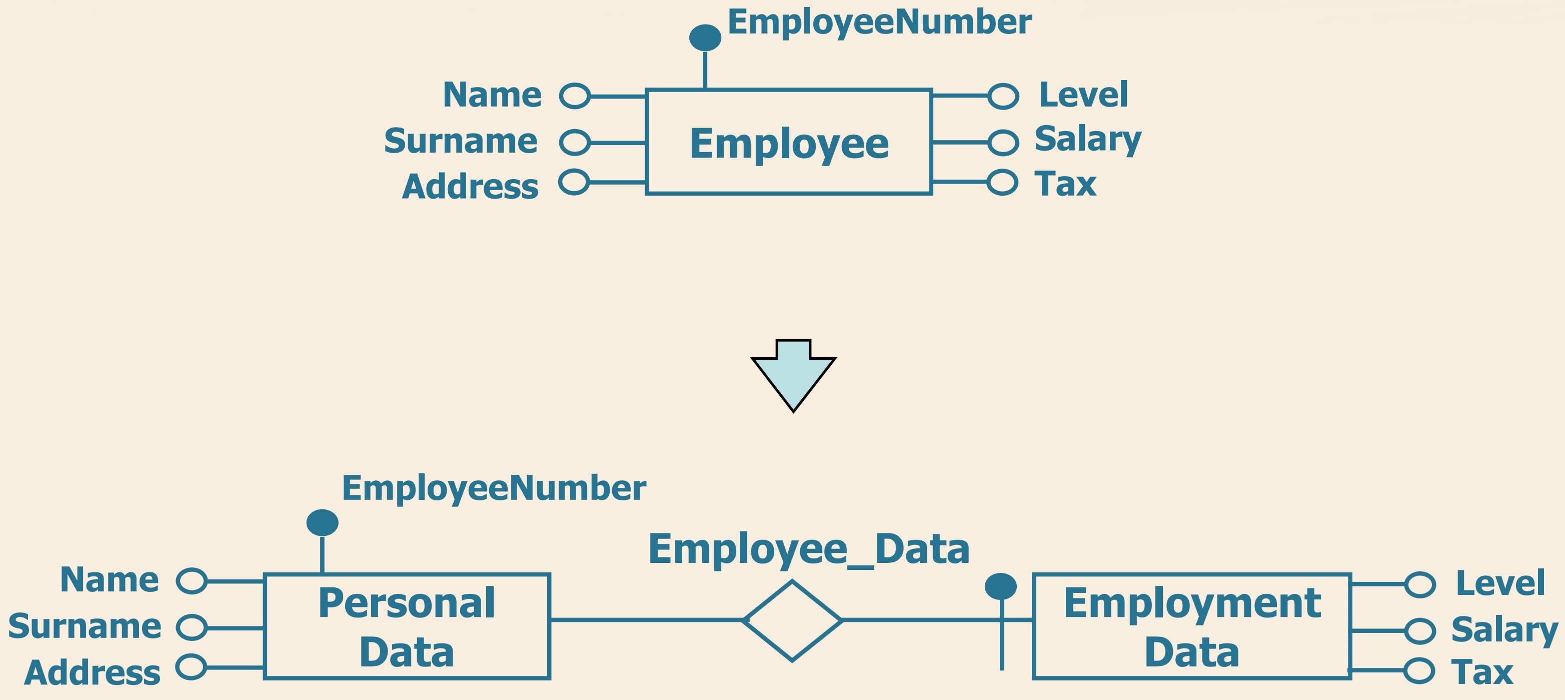
Partitioning of concepts

- Partitioning of entities and relationships
  - Best representation of different concepts
  - Separating attributes of the same concept that are accessed by different operation
  - Improve the efficency of the operations.

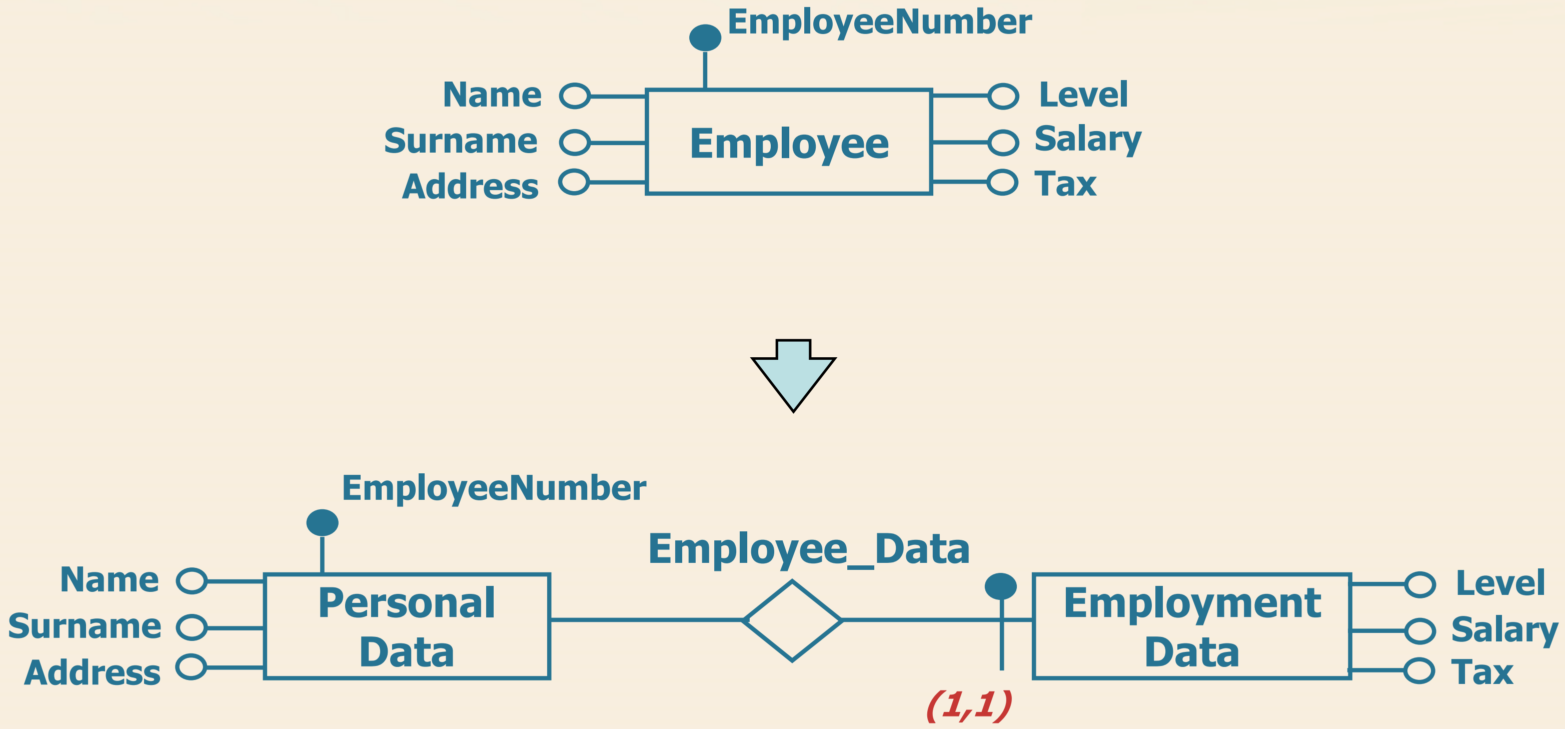
Partizionamento di entità



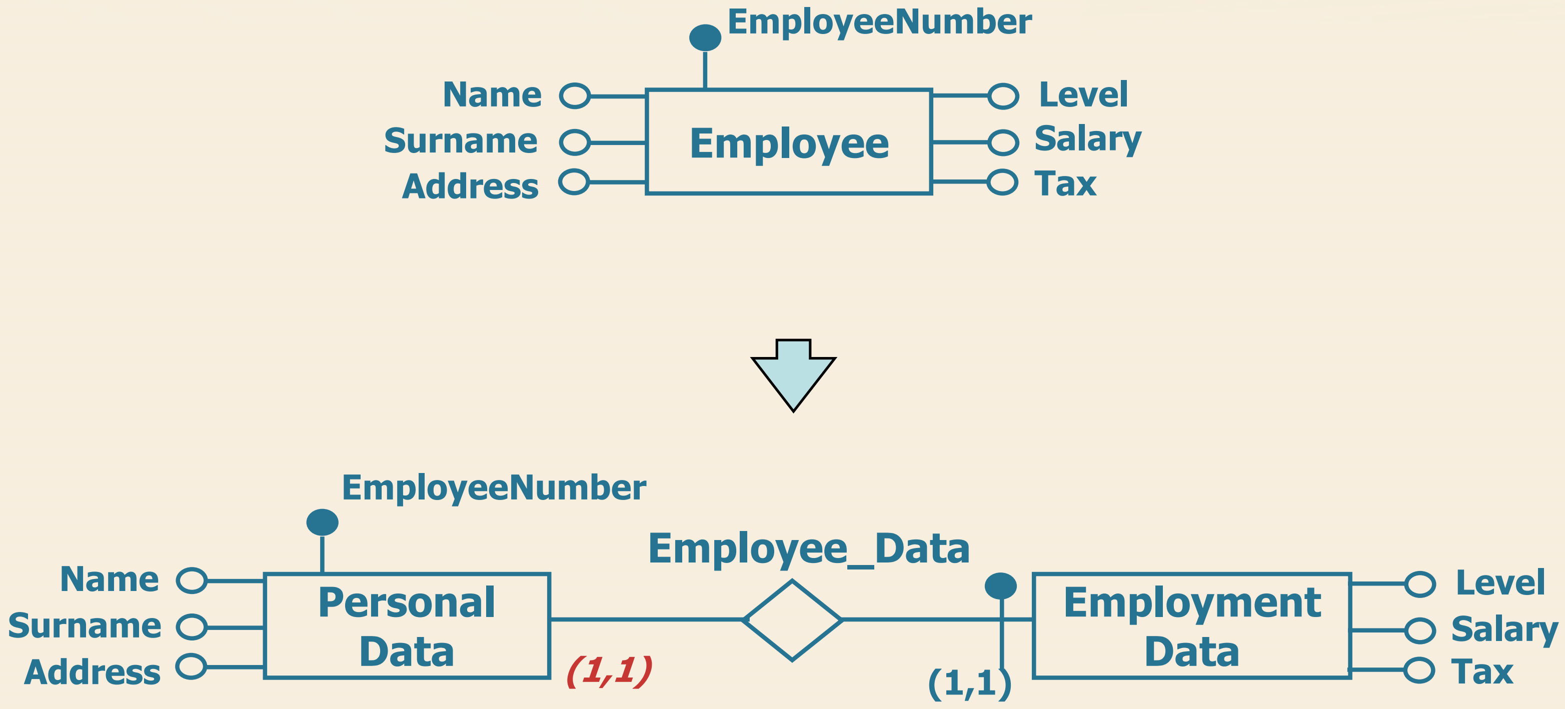
Partizionamento di entità



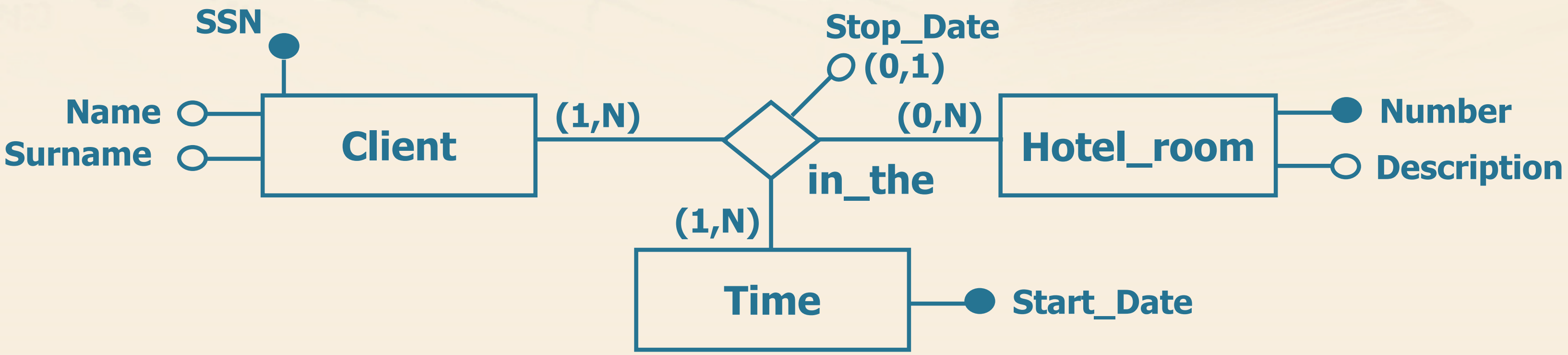
Cardinality of “Employment Data”



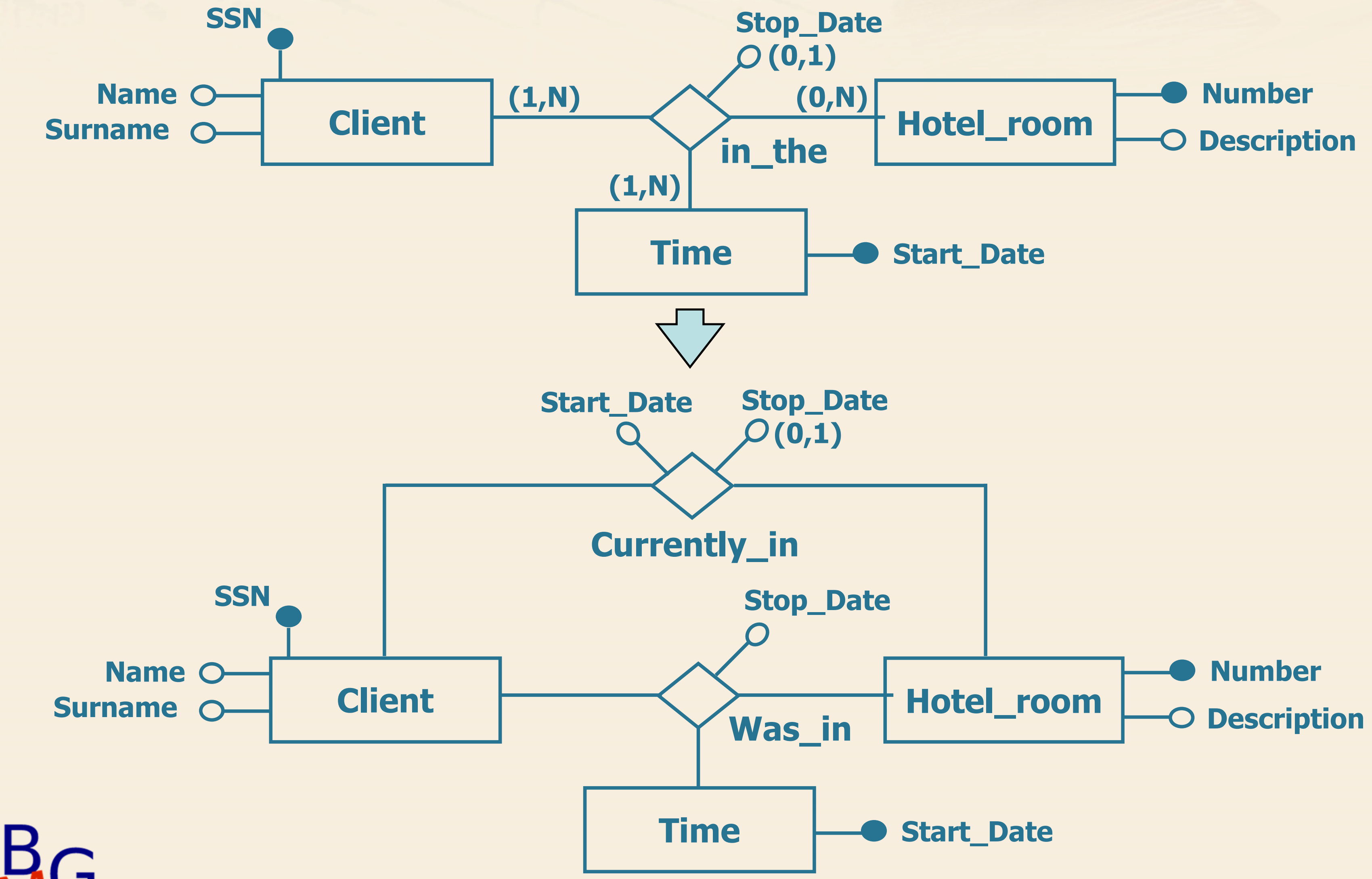
Cardinality of «Employement Data»



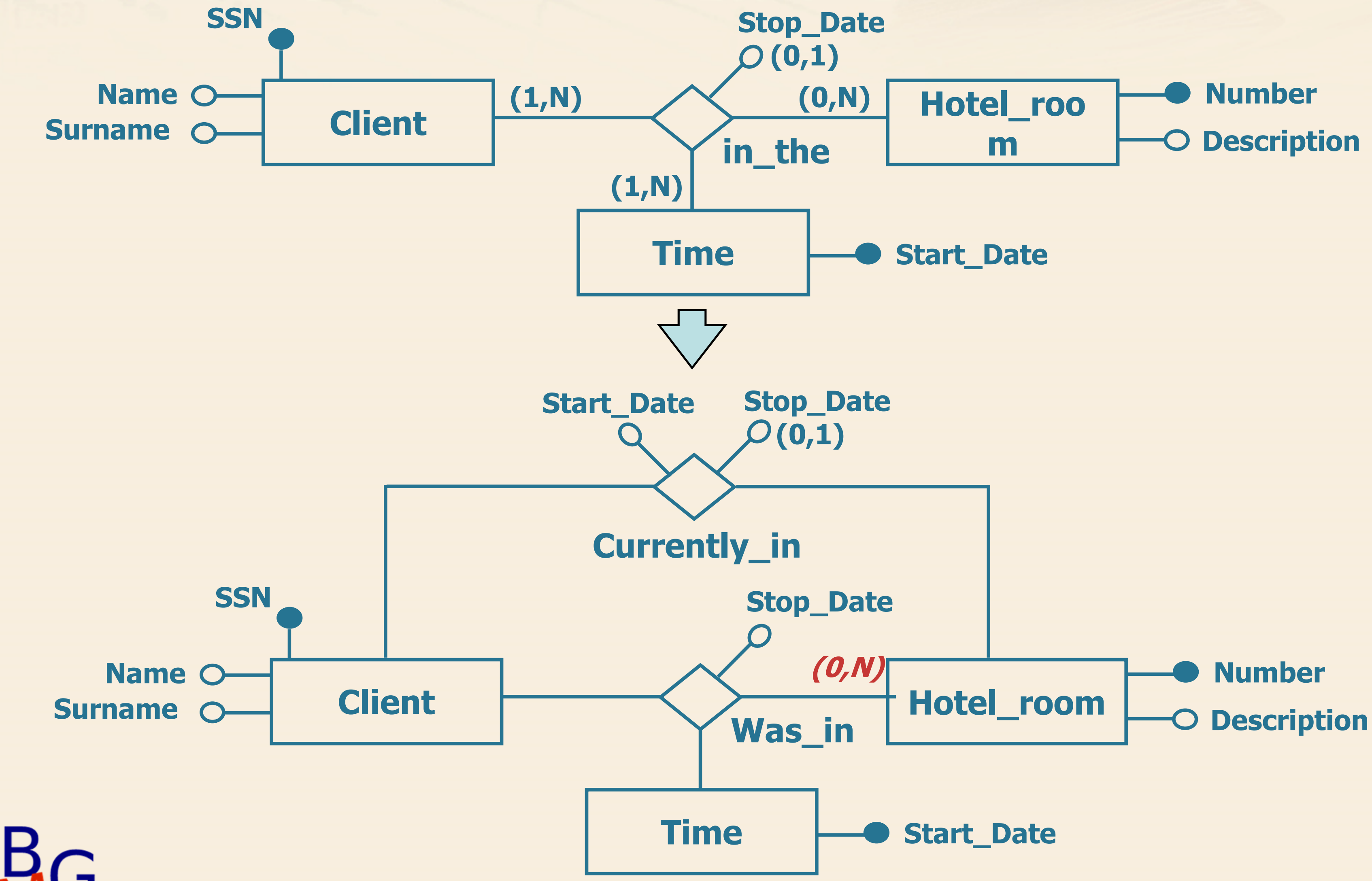
Relationships’ partitioning



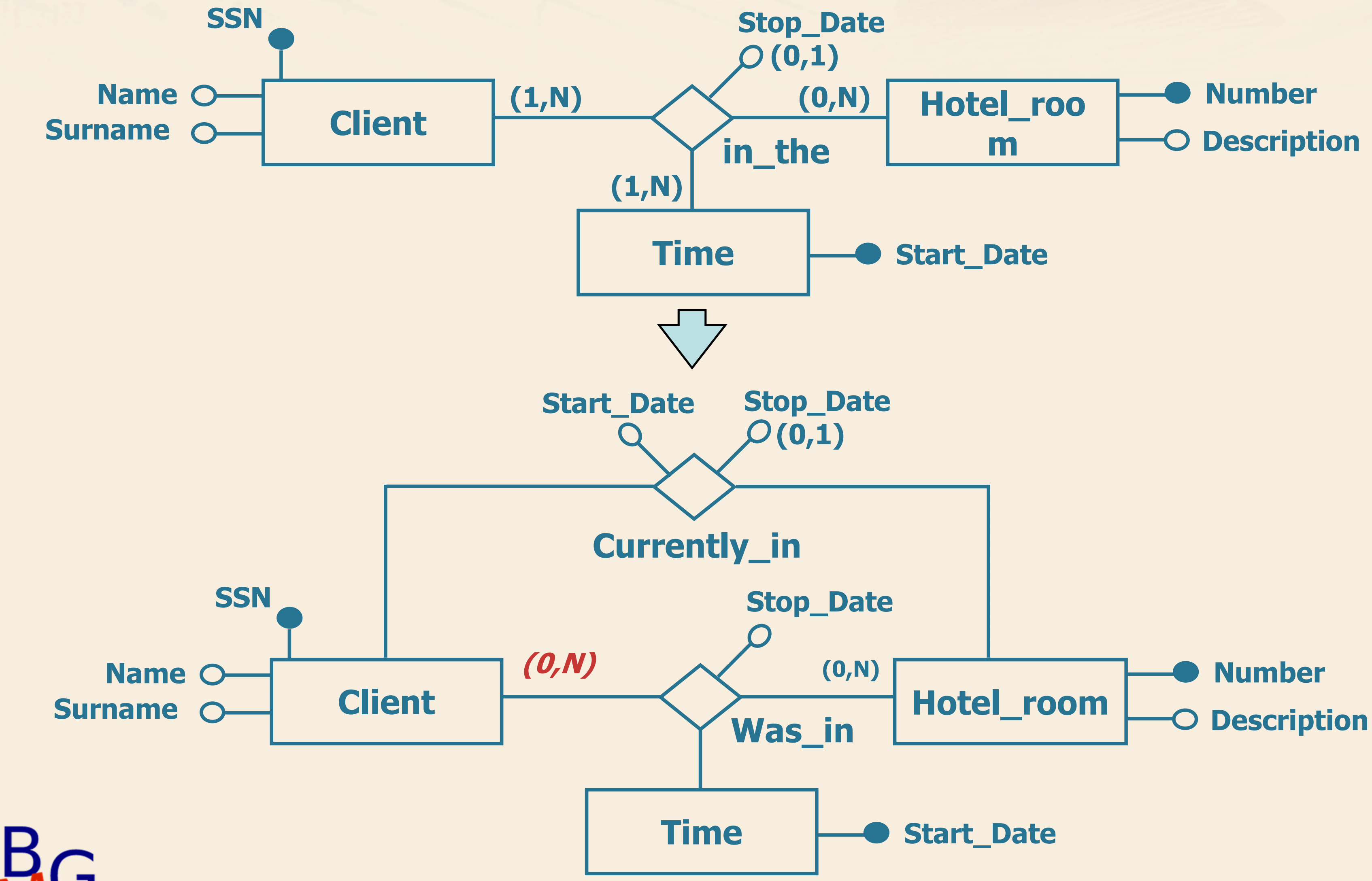
Relationships’ partitioning



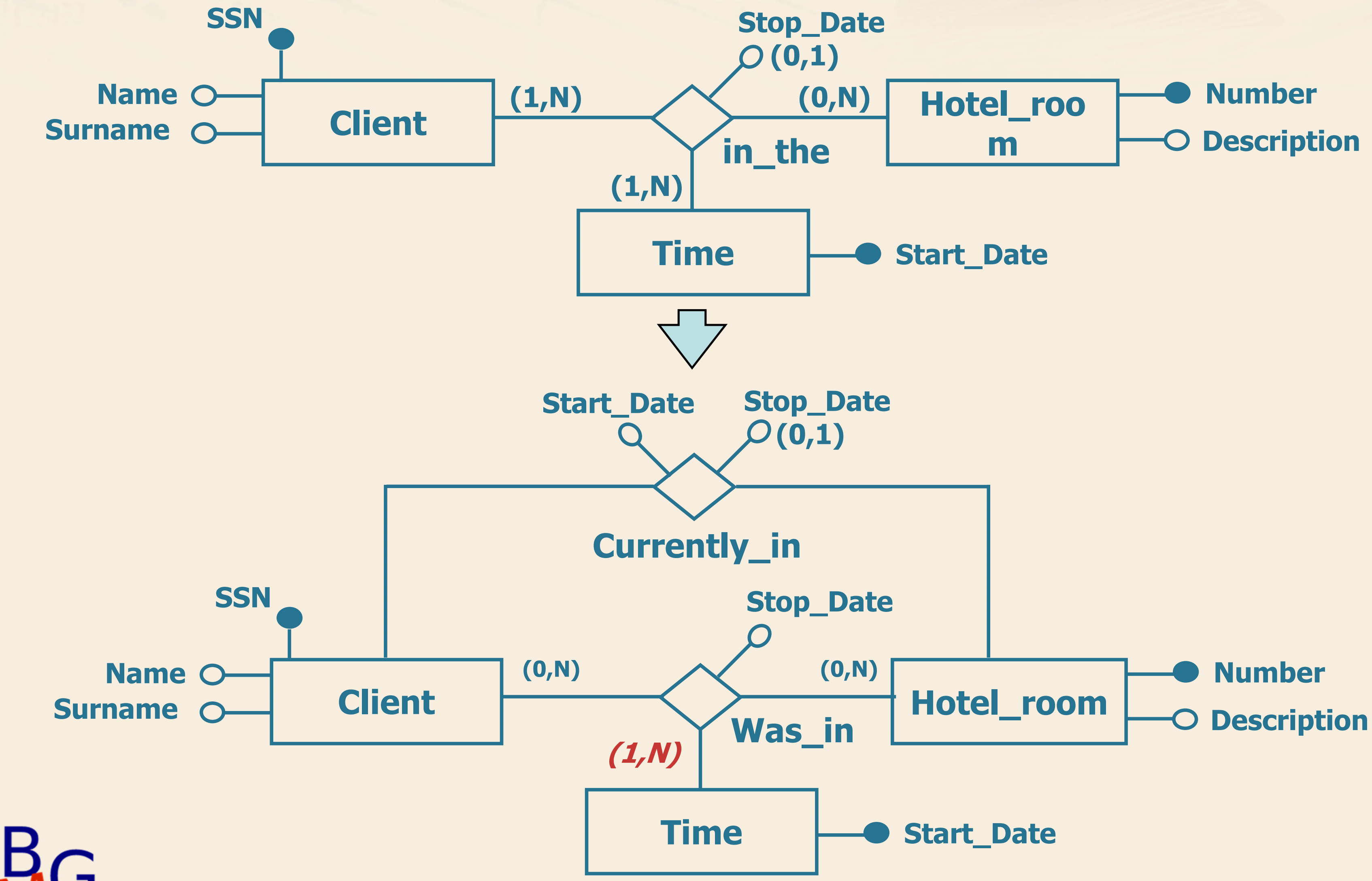
Cardinality of «Was in»



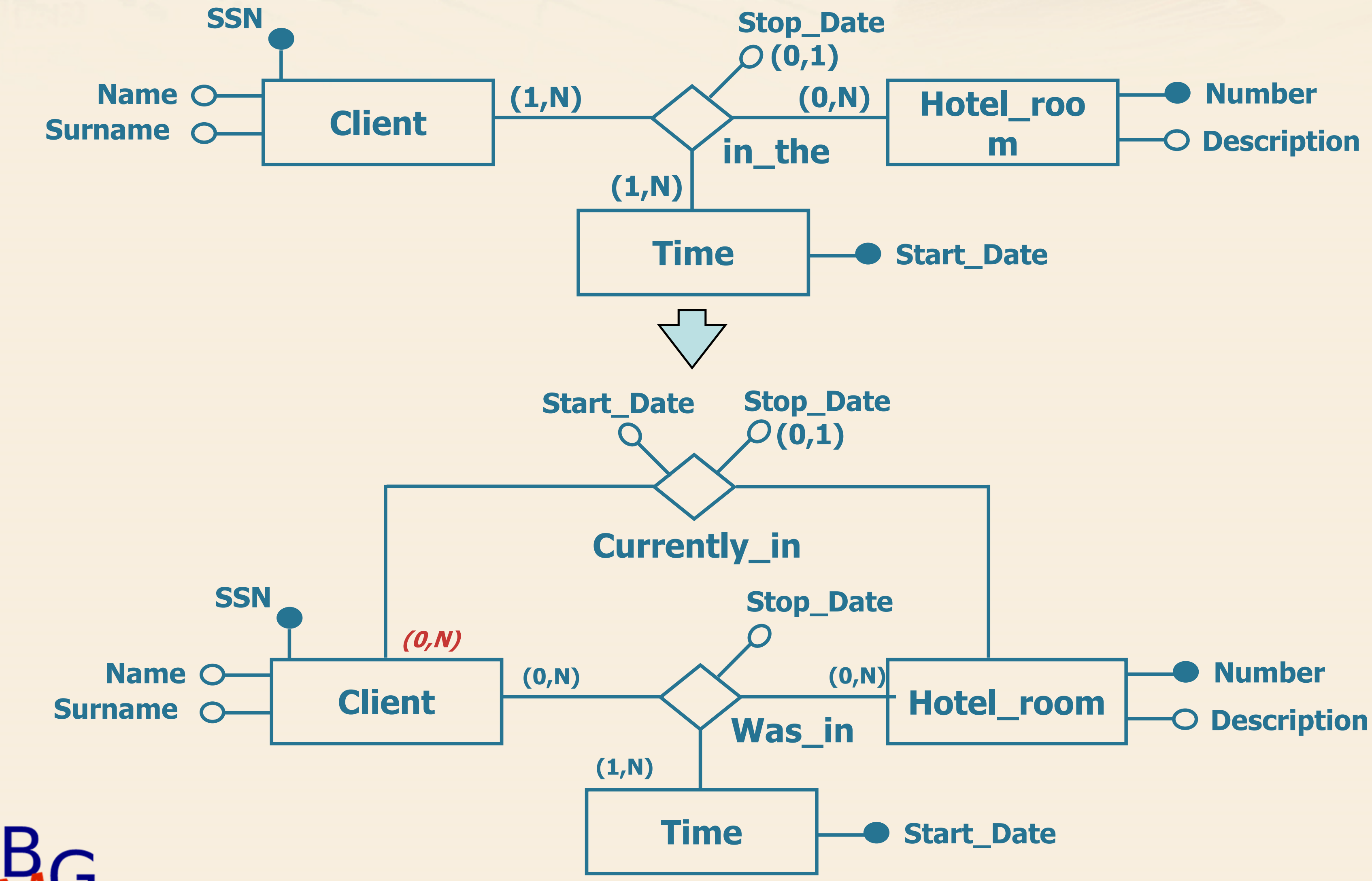
Cardinality of «Was in»



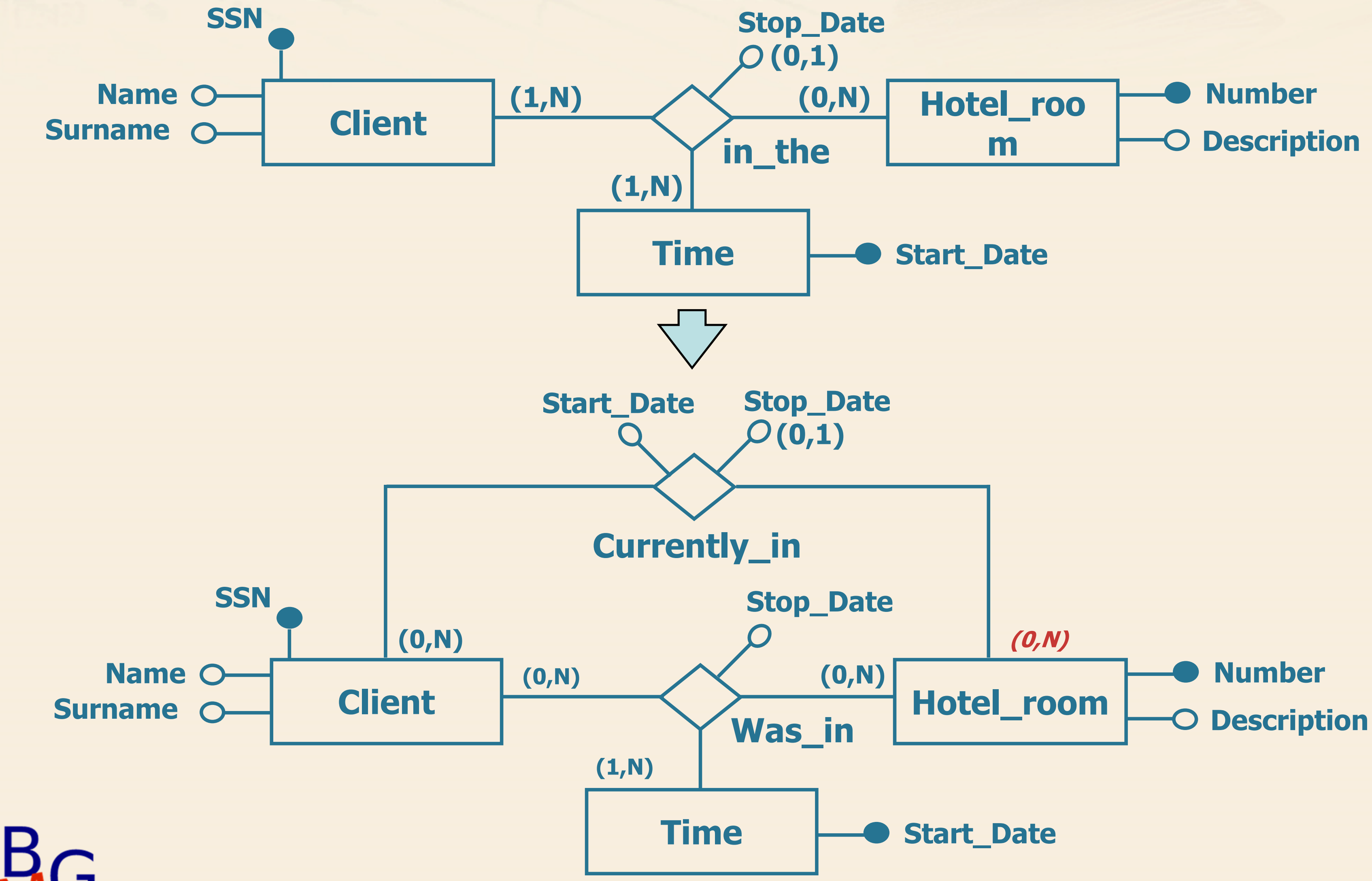
Cardinality of «Was in»



Cardinality of «Currently in»



Cardinality of «Currently in»


SSN
Stop_Date
(0,1)
Name
Surname
Client
(1,N)
(0,N)
Hotel_roo
m
Number
Description
in_the
(1,N)
Time
Start_Date
Start_Date
Stop_Date
(0,1)
Currently_in
SSN
Stop_Date
(0,N)
(0,N)
Name
Surname
Client
(0,N)
(0,N)
Hotel_room
Number
Description
Was_in
(1,N)
Time
Start_Date