Politecnico di Torino

Logical Design

Database Design

Logical Design (1/2)

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

Logical Design (2/2)

- Translation into the relational model
  - entity and many-to-many relationships
  - one-to-many relationships
  - one-to-one relationships
  - entities with external identifiers
  - ternary relationships

Logical Design

Introduction

Logical design

Selection of the logical model

Relational model

Objective

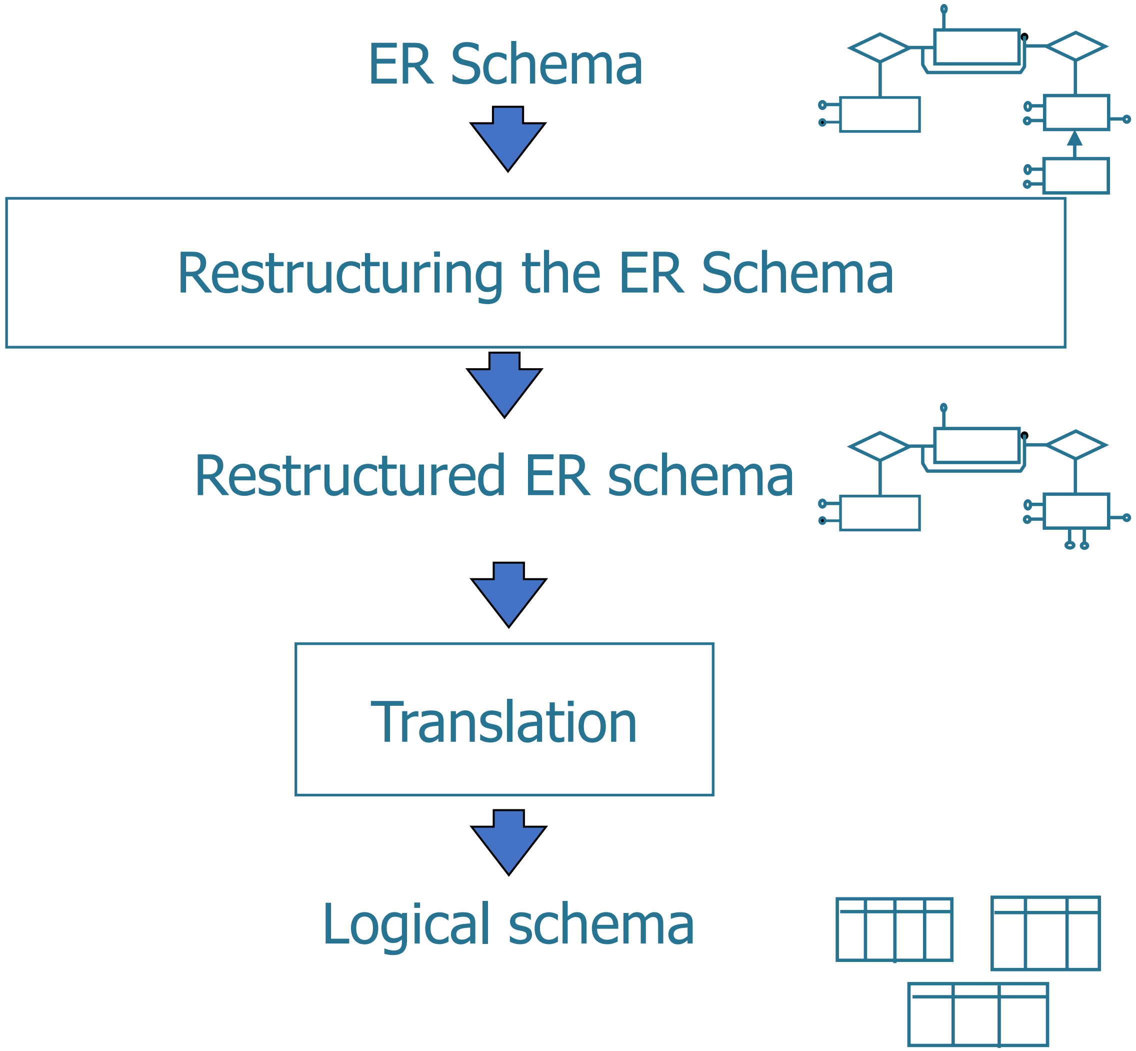
Definition of a relational logical schema corresponding to the starting ER schema

Important

Simplification of the ER schema to make it representable by the relational model

Optimization to increase the efficiency of queries

Logical design steps



ER Schema

Restructuring the ER Schema

Restructured ER schema

Translation

Logical schema

Translation to the relational model

entities and many-to-many relationships

Translation to the relational model

- It is executed on the restructured ER schema
  - i.e., the schema without hierarchies, multivalued attributes and composite attributes
- Transformations
  - Each entity is translated into a table with the same attributes
  - For relationships we need to consider the maximum cardinality

Entity Translation

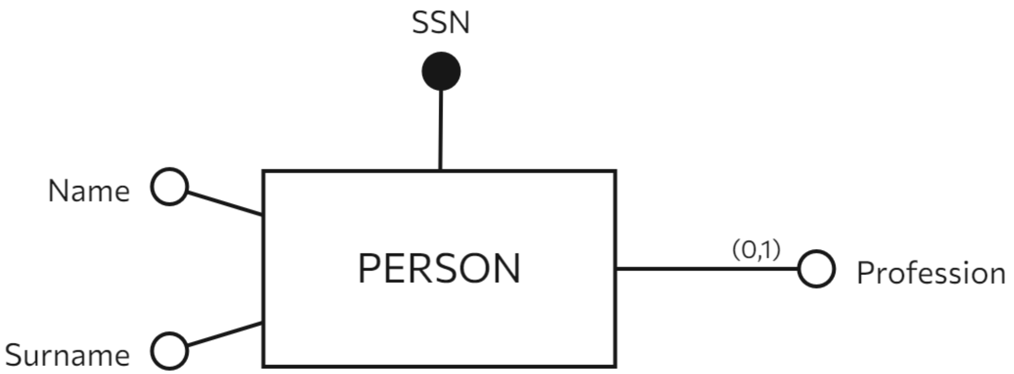
Translating the ER Schema into the Relational Model

Entity Translation

- Each entity corresponds to a table with the same attributes
  - the attributes of the entity constitute the schema of the table
- The identifier (simple or composite) of the Entity becomes the primary key of the table
- Optional Entity attributes are attributes that can be NULL
  - They are highlighted with "*" in the table schema

Entity

Conceptual model



Logical model

Person(<u>SSN</u>, Name, Surname, Profession*)

- Underlined primary key
- Optional attributes indicated with an asterisk

Relationship translation

Translating the ER Schema into the Relational Logic Model

Relationship translation

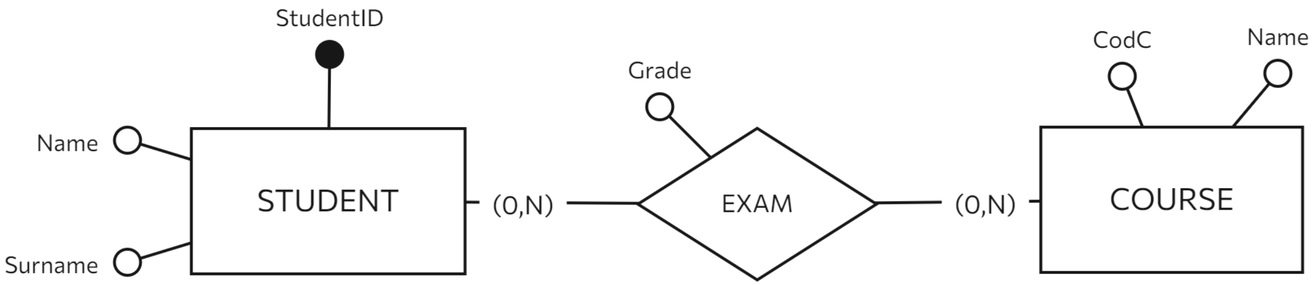
- To translate a relationship
  1. Step 1: The Entities participating in the Relationship are first translated
  2. Step 2: The Relationship is then translated
     - Different translation rules for binary and ternary Relationships
     - For a Binary Relationship, it is necessary to consider the maximum and minimum cardinality with which the Entities participate in the Relationship

Translation of Binary Relationships

Translating the ER Schema into the Relational Model

Many-to-many binary relationship

Conceptual model



Logical model

Student(<u>StudentID</u>, Name, Surname)

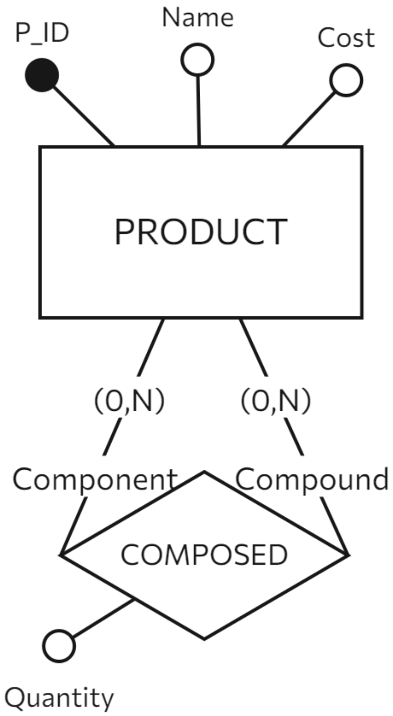
Course(<u>CodC</u>, Name)

Exam(<u>StudentID</u>, <u>CodC</u>, Grade)

- Each many-to-many relationship corresponds to a table
  - The primary key is the combination of the identifiers of the two linked entities
  - The attributes of the table that corresponds to the relationship can be renamed (required in case of recursive relationships)

Recursive many-to-many binary relationship

Conceptual model



Logical model

Product (<u>P_ID</u>, Name, Cost)

Composed(<u>CodCompound</u>, <u>CodComponent</u>, Quantity)

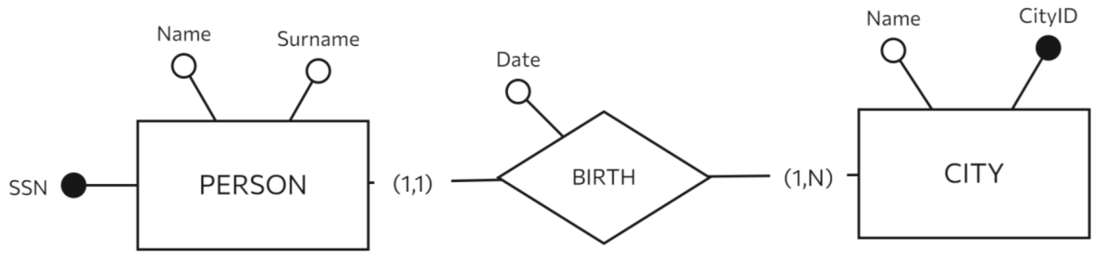
- Each many-to-many relationship corresponds to a table
  - The primary key is the combination of the identifiers of the two linked entities
  - The attributes of the table that corresponds to the relationship can be renamed (required in case of recursive relationships)

One-to-many binary relationship

- Two translation modes are possible
  - by means of attributes
  - by means of a new table

One-to-Many Binary Relationship: using attributes

Conceptual model



Logical model

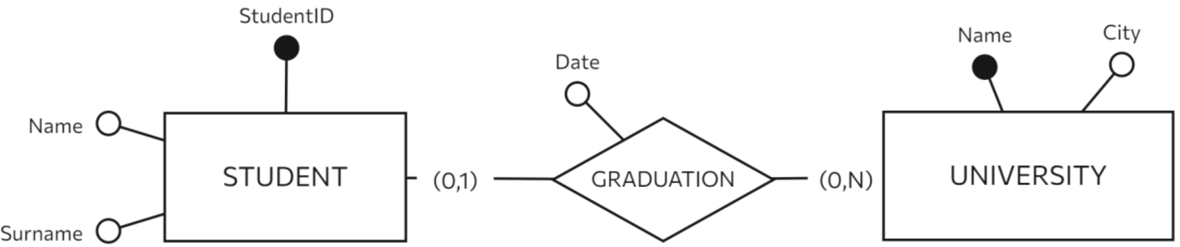
Person (SSN, Name, Surname, CityID, Date)

City (CityID, Name)

- It is used when participation of the entity that participates with a maximum cardinality of 1 is mandatory (minimum cardinality of 1)

One-to-many binary relationship: using attributes or a new table

Conceptual model



Logical model

Alternative 1: Translation using attributes

Student (StudentID, Name, Surname, NameUniv*, Date*)

University (Name, City)

Alternative 2: Translation using a new table

Student (StudentID, Name, Surname)

University (Name, City)

Graduation (StudentID, NameUniv, Date)

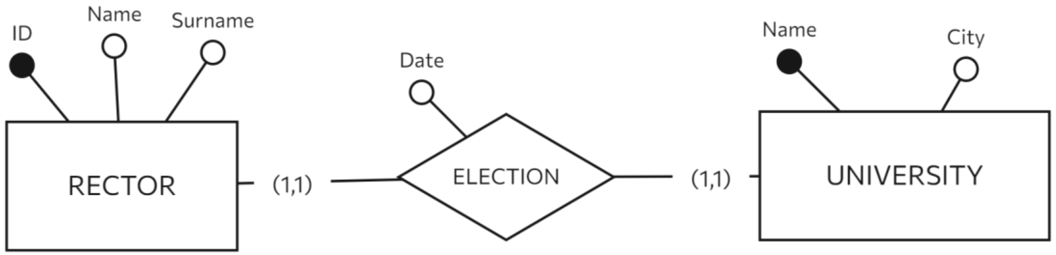
- It is used when participation of the entity that participates with a maximum cardinality of 1 is optional (minimum cardinality of 0)

One-to-one binary relationship

- Multiple translations are possible
  - depends on the value of the minimum cardinality

One-to-one binary relationship

Conceptual model



Logical model

Alternative 1

Rector (ID, Name, Surname, UnivName, Date)

University (Name, City)

Alternative 2

Rector (ID, Name, Surname)

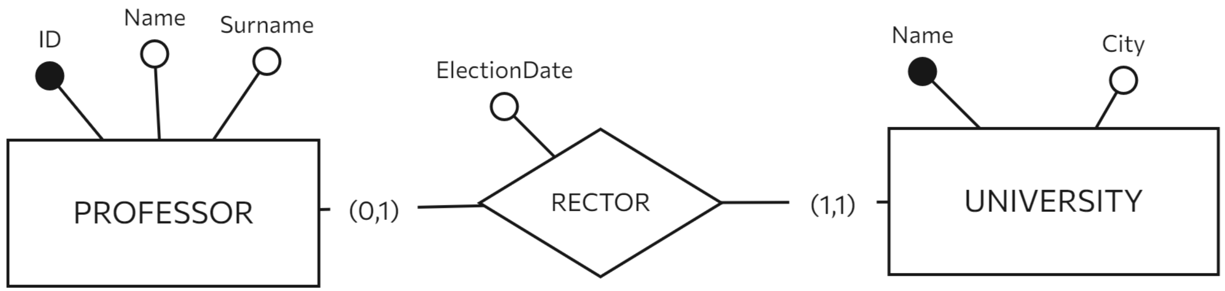
University (Name, City, ID, Date)

- It is used when both entities participate with a maximum cardinality of 1 in the relationship, and participation is mandatory for both entities (minimum cardinality of 1)

One-to-one binary relationship

Conceptual model

Logical model



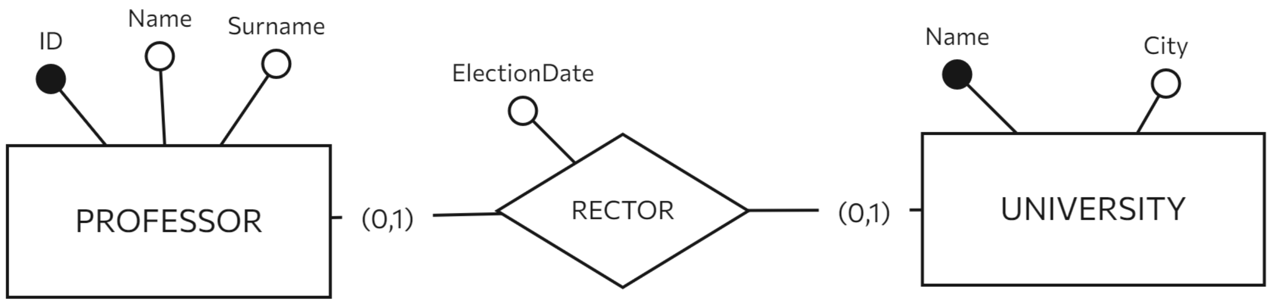
Professor (<u>ID</u>, Name, Surname)

University (<u>Name</u>, City, RectorID, ElectionDate)

- It is used when both entities participate with a maximum cardinality of 1 in the relationship, but participation is mandatory only for one entities (minimum cardinality of 1)

One-to-one binary relationship

Conceptual model



Logical model

Alternative 1

Professor (ID, Name, Surname)
University (Name, City)
Rector (RectorID, UniversityName, ElectionDate)

Alternative 2

Professor (ID, Name, Surname)
University (Name, City)
Rector (RectorID, UniversityName, ElectionDate)

Alternative 3

Professor (ID, Name, Surname)
University (Name, City, RectorID*, ElectionDate*)

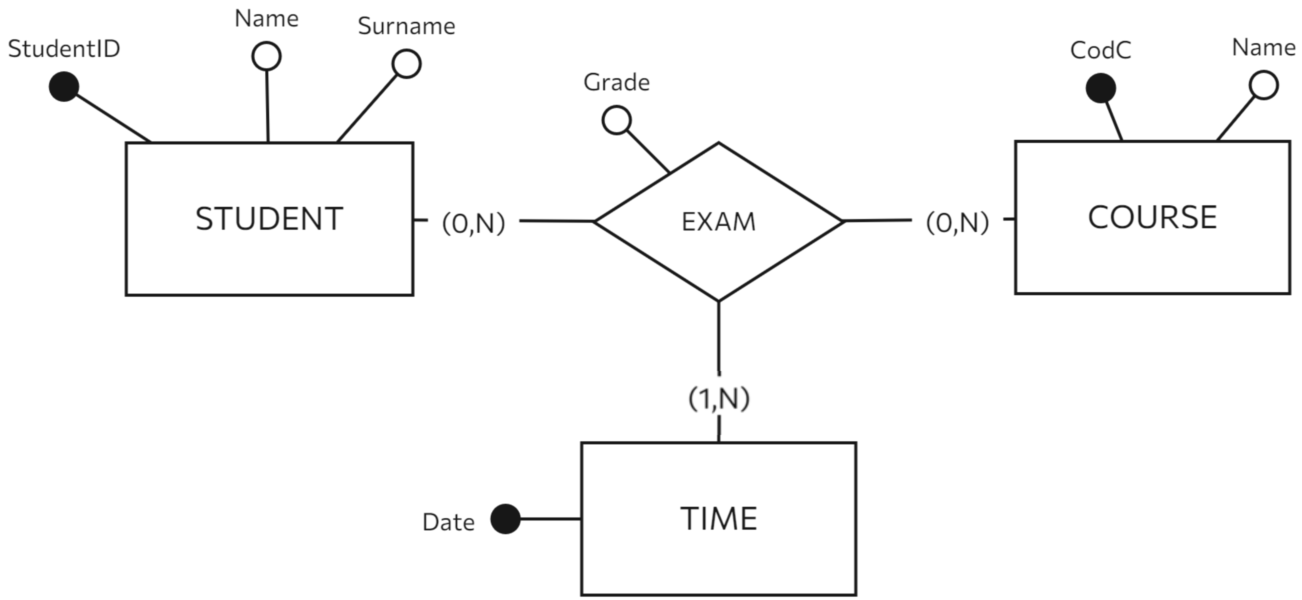
- It is used when both entities participate with a maximum cardinality of 1 in the relationship, and participation is optional for both entities (minimum cardinality of 0)

Translation of Ternary Relationships

Translating the ER Schema into the Relational Model

Ternary Relationship

Conceptual model



Logical model

Student(<u>StudentID</u>, Name, Surname)

Course(<u>CodC</u>, Name)

Time(<u>Date</u>)

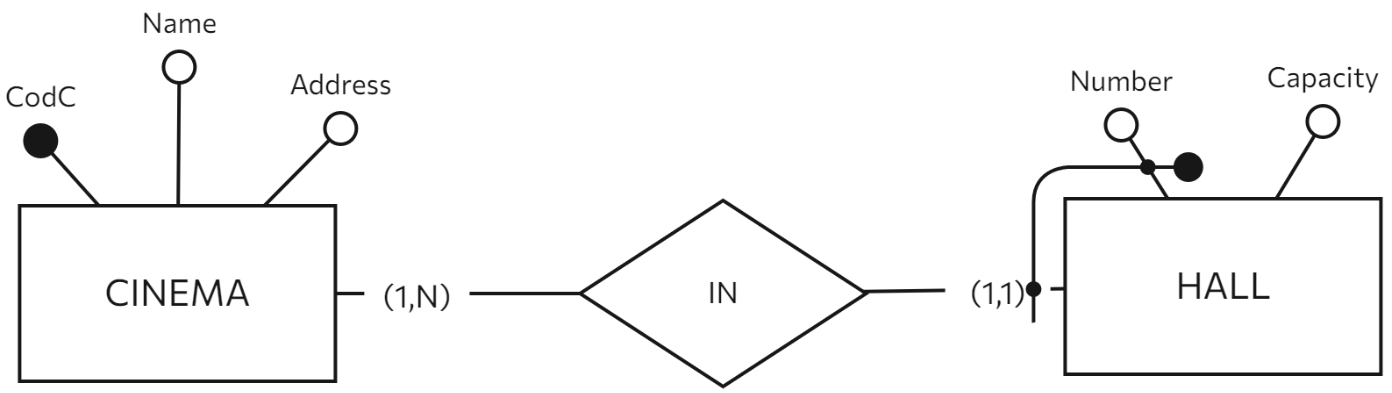
Exam(<u>StudentID</u>, <u>CodC</u>, <u>Date</u>, Grade)

Translating Entities with External Identifier

Translating the ER Schema into the Relational Model

Entities with an external identifier

Conceptual model



Logical model

Cinema (<u>CodC</u>, Name, Address)

Hall (<u>Number</u>, <u>CodC</u>, Capacity)

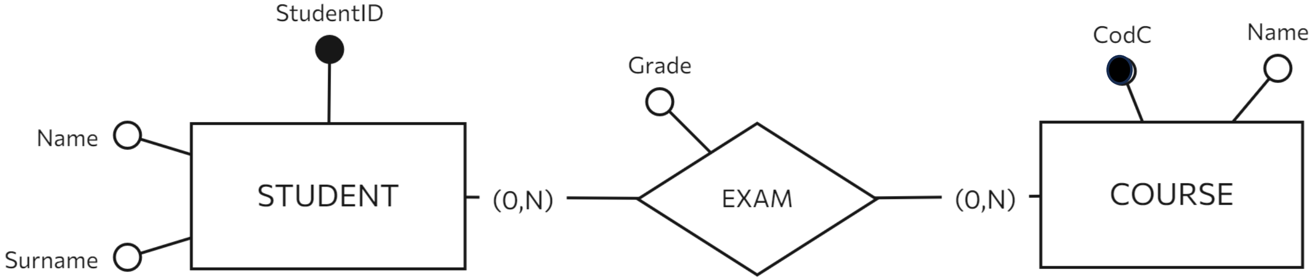
- The relationship is represented together with the identifier
- The relationship contributes to the definition of the weak entity identifier

Referential integrity constraints

Translating the ER Schema into the Relational Model

Referential integrity constraints

Conceptual model



Logical model

Student(<u>StudentID</u>, Name, Surname)

Course(<u>CodC</u>, Name)

Exam(<u>StudentId</u>, <u>CodC</u>, Grade)

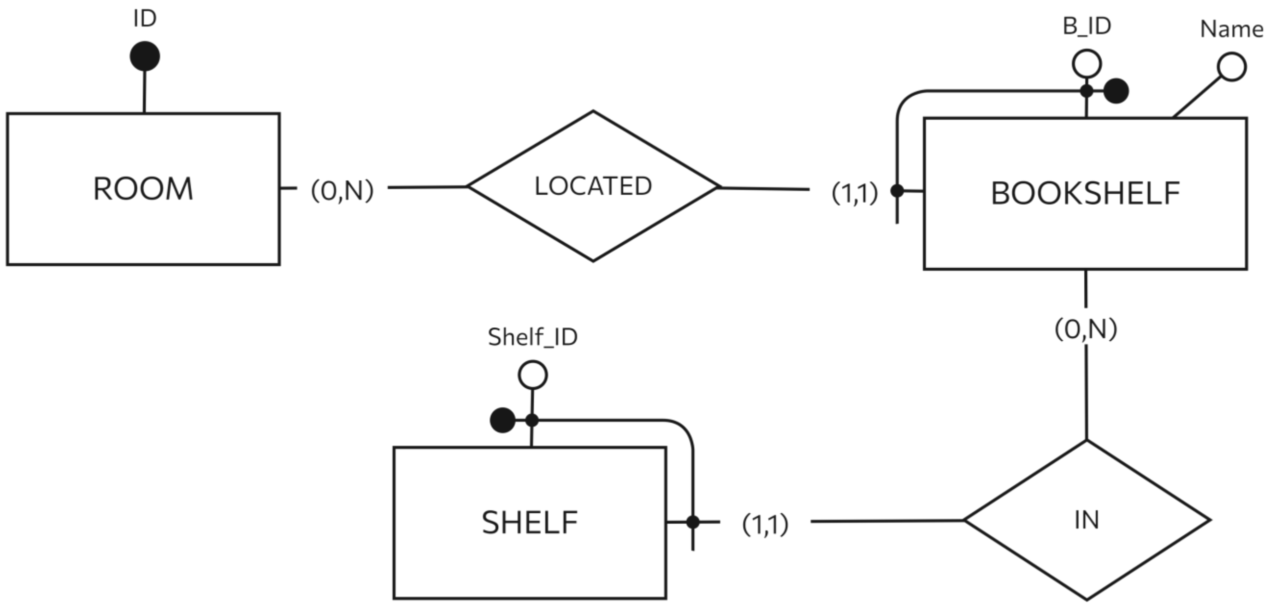
- Relationships Represent Referential Integrity Constraints

Exam(StudentID) REFERENCES Student(StudentID)

Exam(CodC) REFERENCES Course(CodC)

Referential integrity constraints

Conceptual model



Logical model

Room (ID)

Bookshelf (*ID*, B_ID, Name)

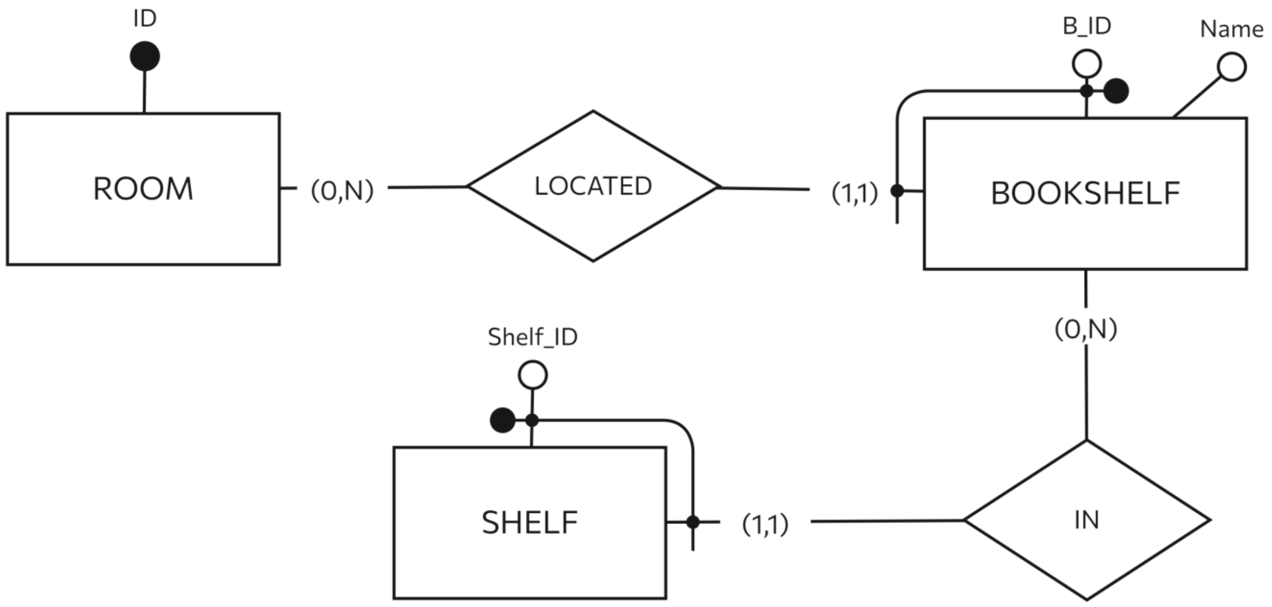
Shelf (*ID*, *B_ID*, Shelf_ID)

- If the referenced key consists of multiple attributes, the referential integrity constraint is imposed on the attribute set

  Bookshelf(ID) REFERENCES Room(ID)
  Shelf (ID, B_ID) REFERENCES Bookshelf (ID, B_ID)

Referential integrity constraints

Conceptual model



Logical model

Room (ID)

Bookshelf (*ID*, B_ID, Name)

Shelf (*ID*, *B_ID*, Shelf_ID)

- If the referenced key consists of multiple attributes, the referential integrity constraint is imposed on the attribute set

Bookshelf(ID) REFERENCES Room(ID)
~~Shelf (ID) REFERENCES Bookshelf (ID)~~
~~Shelf(B_ID) REFERENCES Bookshelf (B_ID)~~

Wrong constraints!

Restructuring the ER model

Restructuring the ER model

Restructuring the ER model

- The restructured ER model takes into account implementation aspects
  - It is no longer a conceptual model
- Objectives
  - To eliminate constructs for which there is no direct representation in the relational model
  - To transform the data representation in order to increase the efficiency of data access operations

Restructuring tasks

- Eliminating composite attributes
- Eliminating multivalued attributes
- Eliminating generalizations
- Analysis of redundancies
- Partitioning concepts (Entities, Relationships)
- Choosing primary identifiers

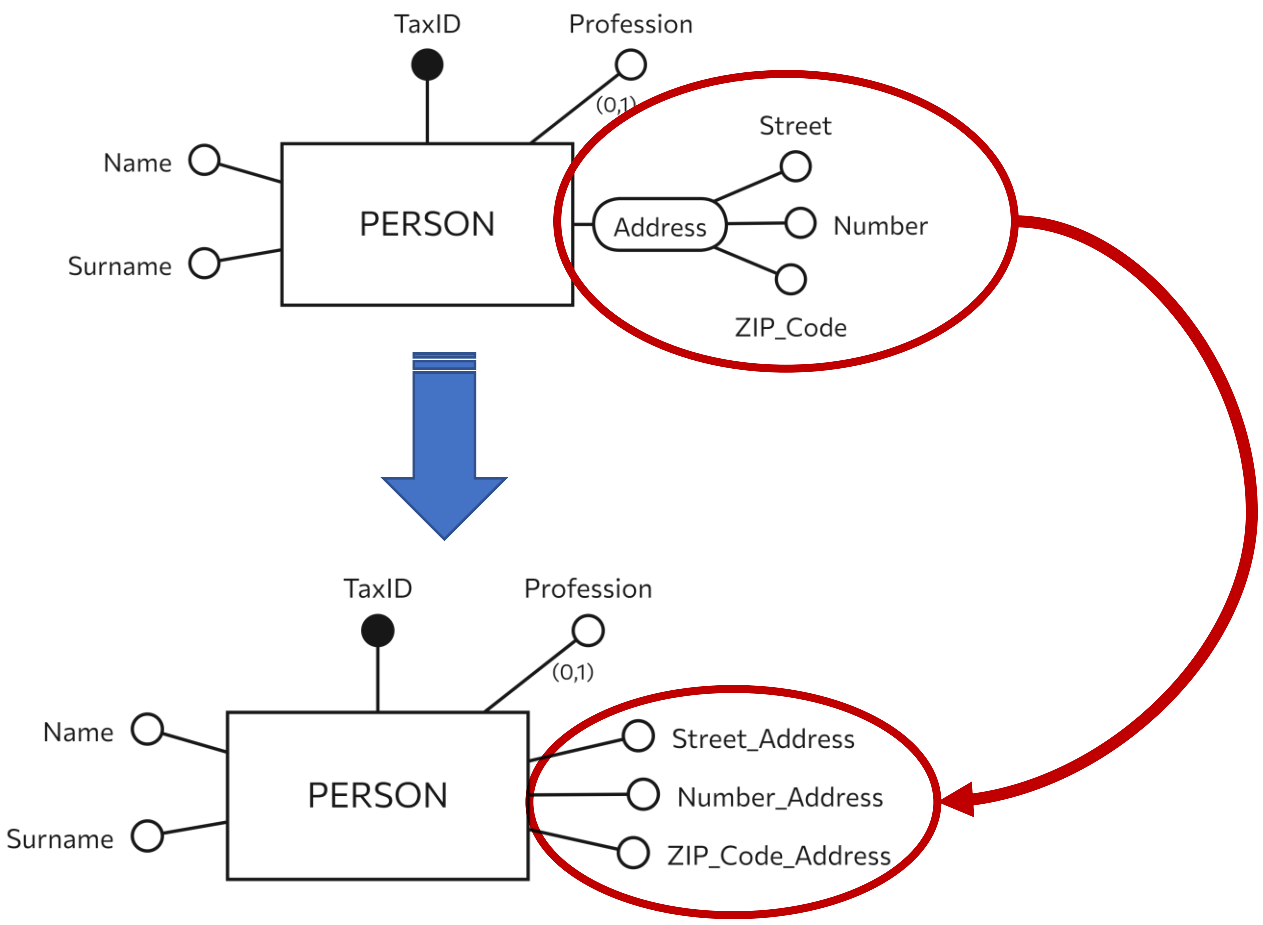
Eliminating composite attributes

Restructuring the ER model

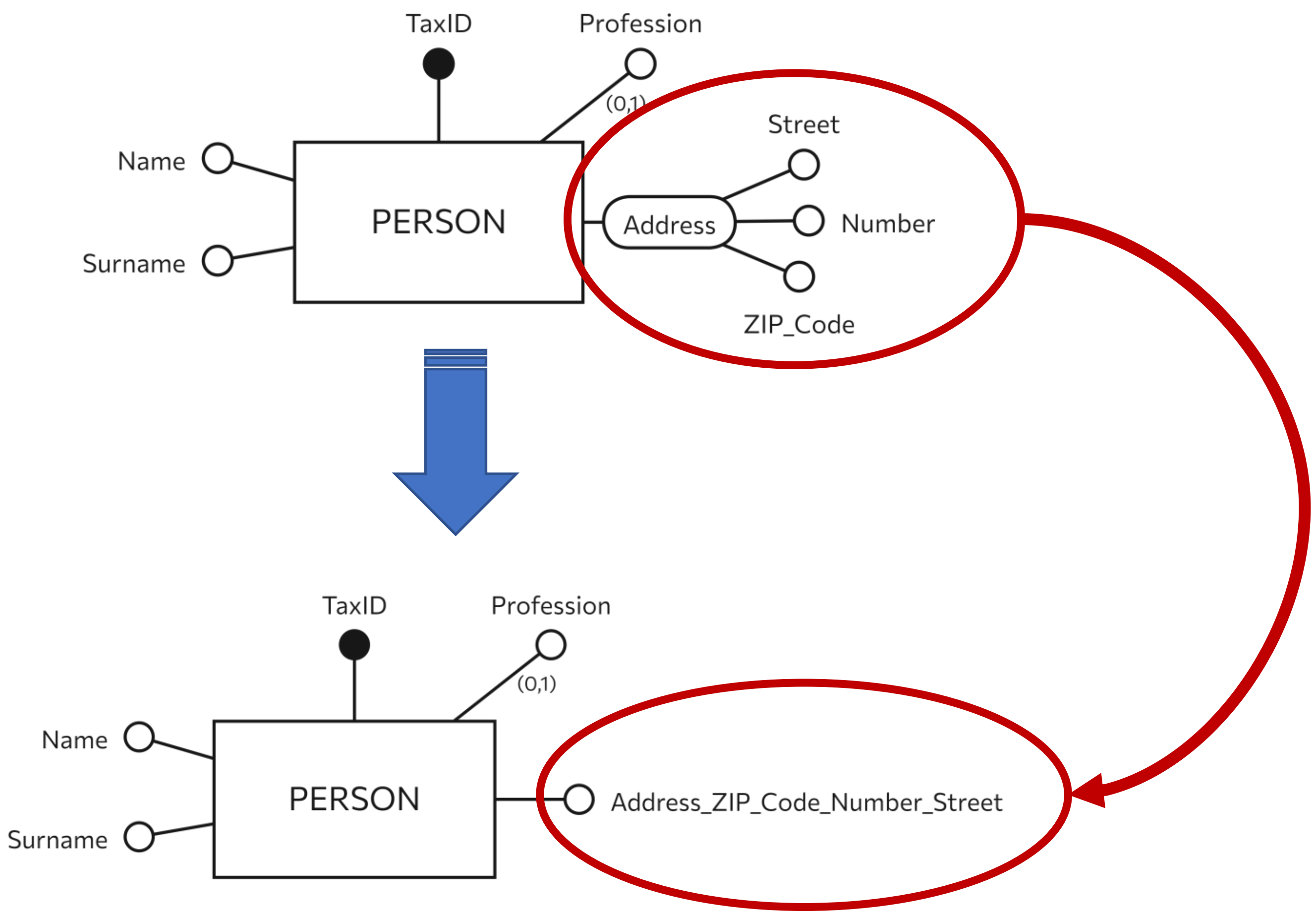
Eliminating composite attributes

- Composite attributes are not representable in the relational model.
- Attributes can be deleted by:
  - **separately representing individual sub-attributes**
    - if you need to access each attribute separately
  - **Introducing a single attribute that represents the concatenation of the composite attributes**
    - if access to the overall information is sufficient

Option 1: separate attributes



Option 2: single attribute



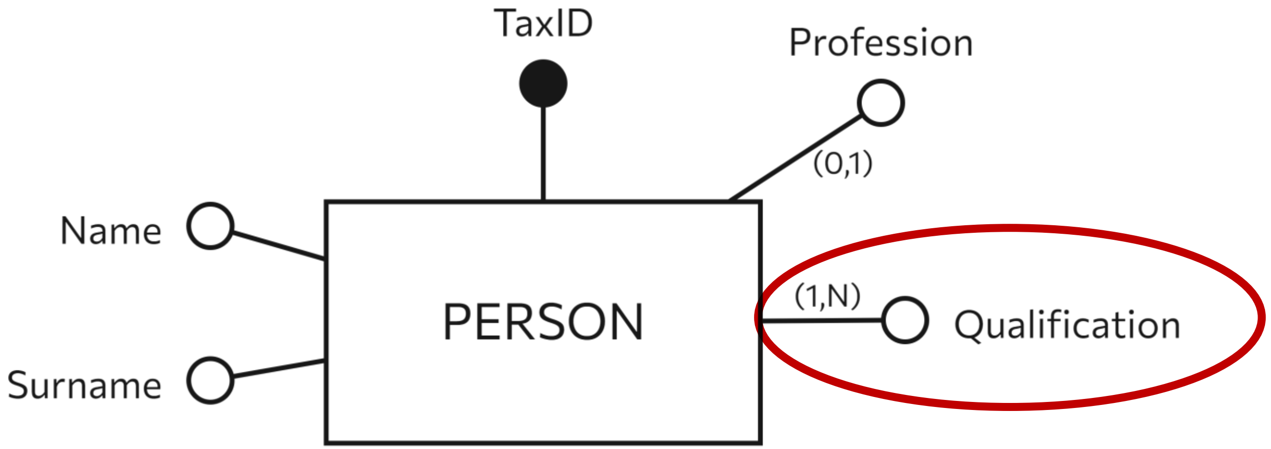
Eliminating multivalued attributes

Restructuring the ER model

Eliminating multivalued attributes

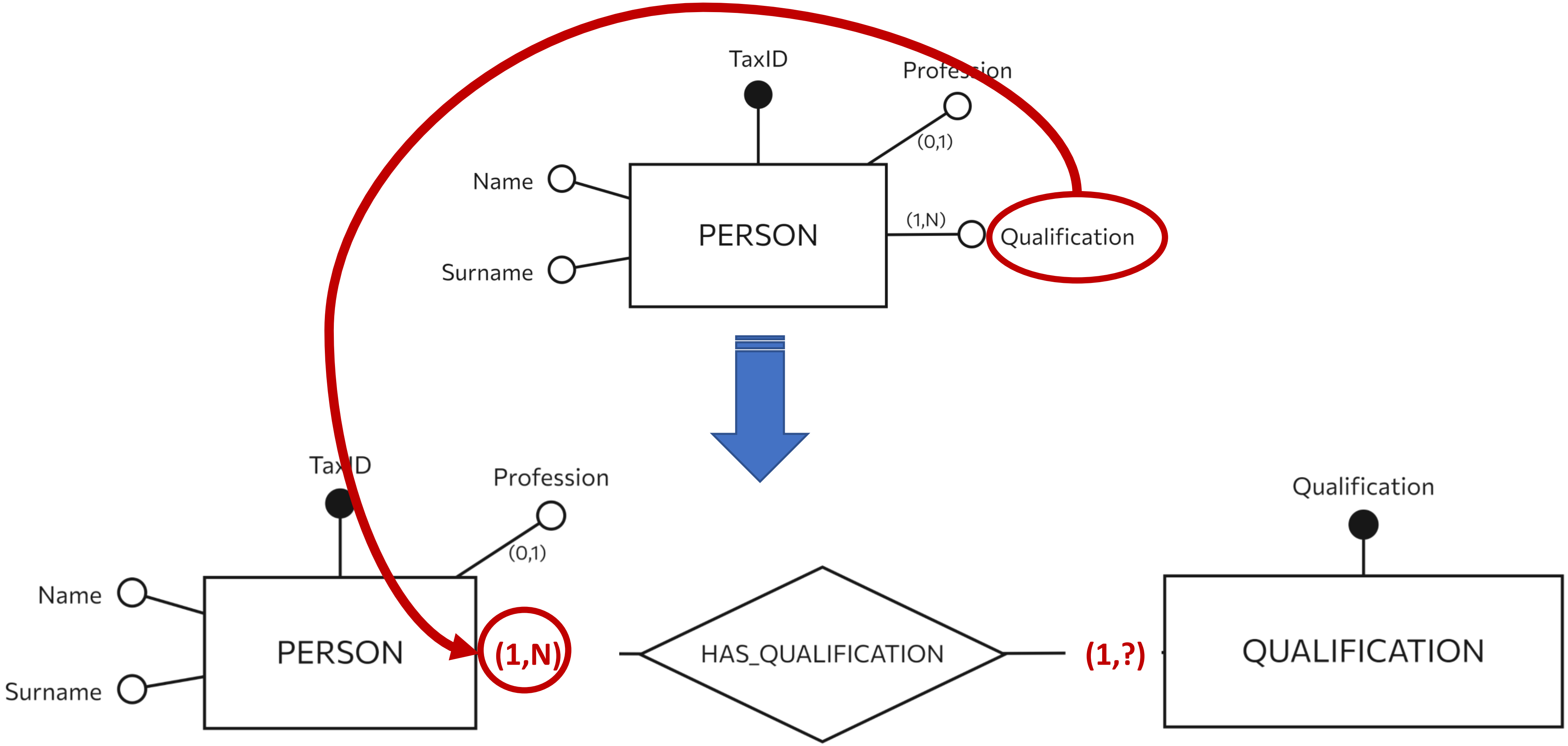
- They cannot be represented in the relational model
- Multivalued attributes are represented using a relationship between:
  - the initial entity
  - a new entity
- Pay attention to the cardinality of the new relationship

Shared information

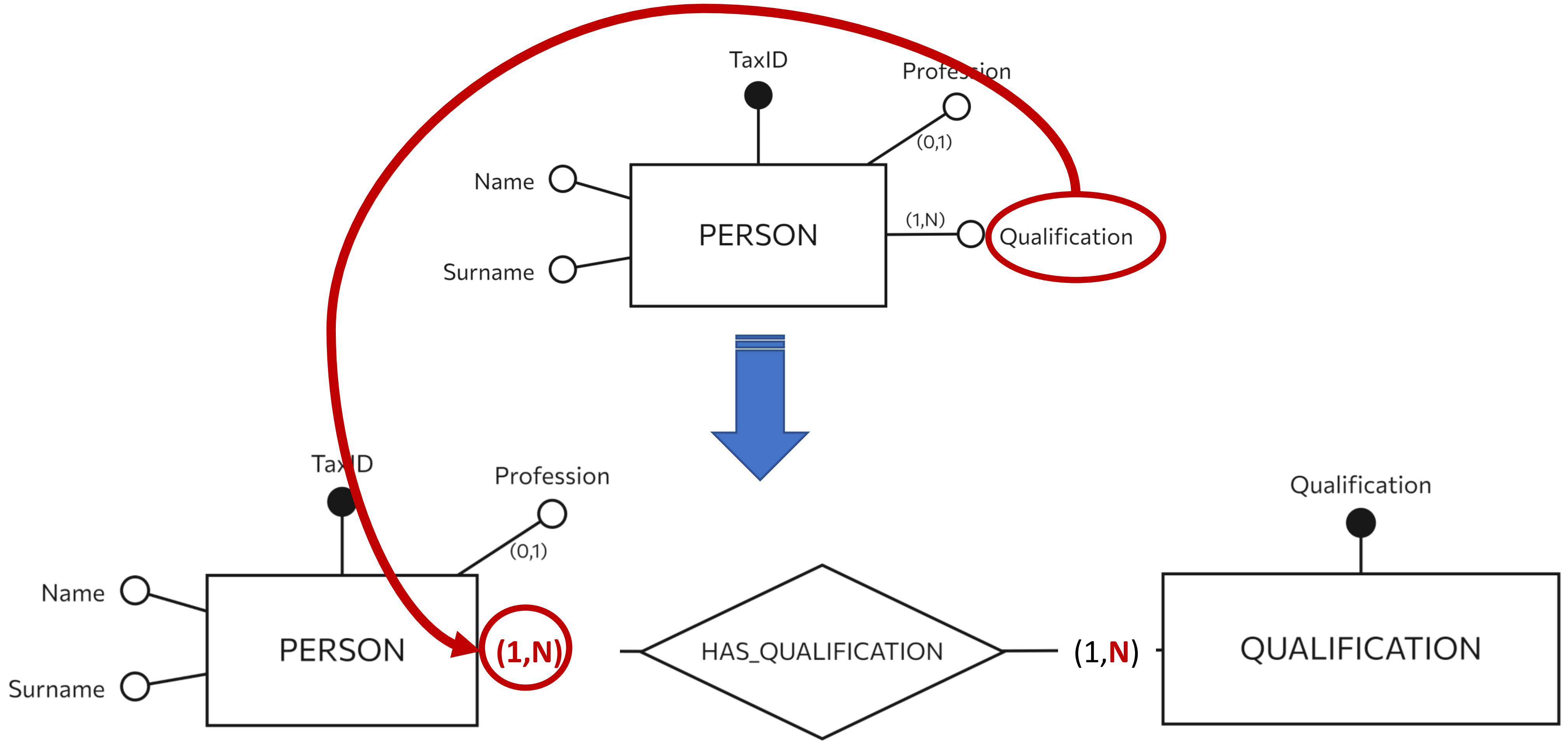


A person can have more than one educational qualification and that the same educational qualification can be held by several people

Shared information: *Has_qualification* cardinality

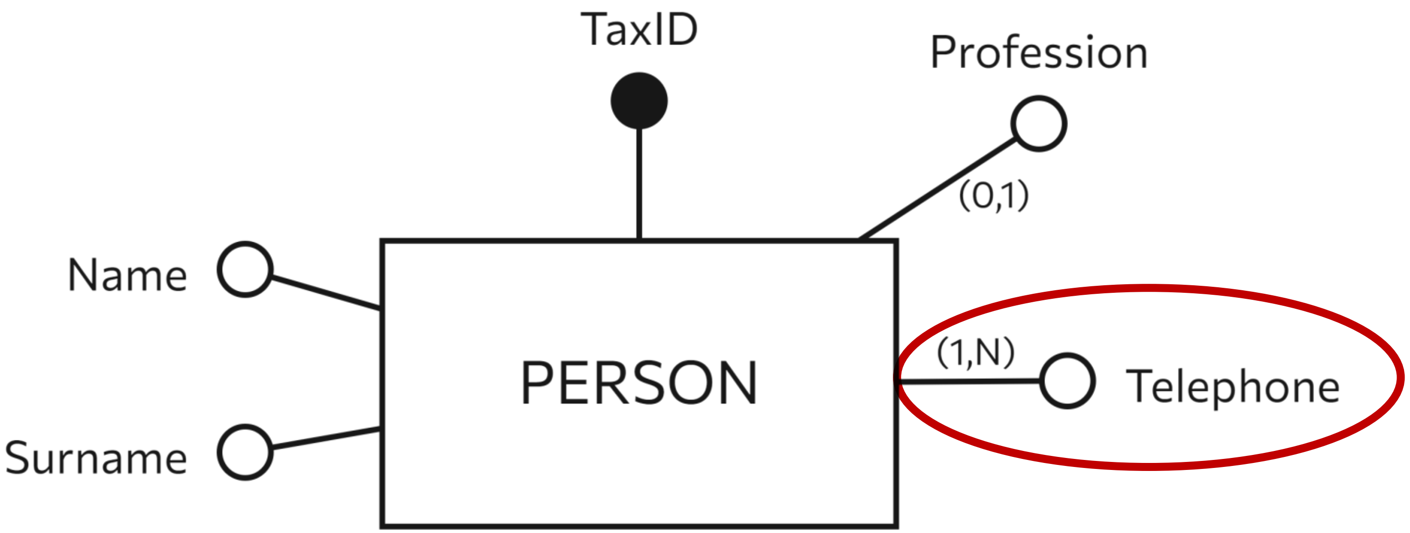


Shared information: *Has_qualification* cardinality



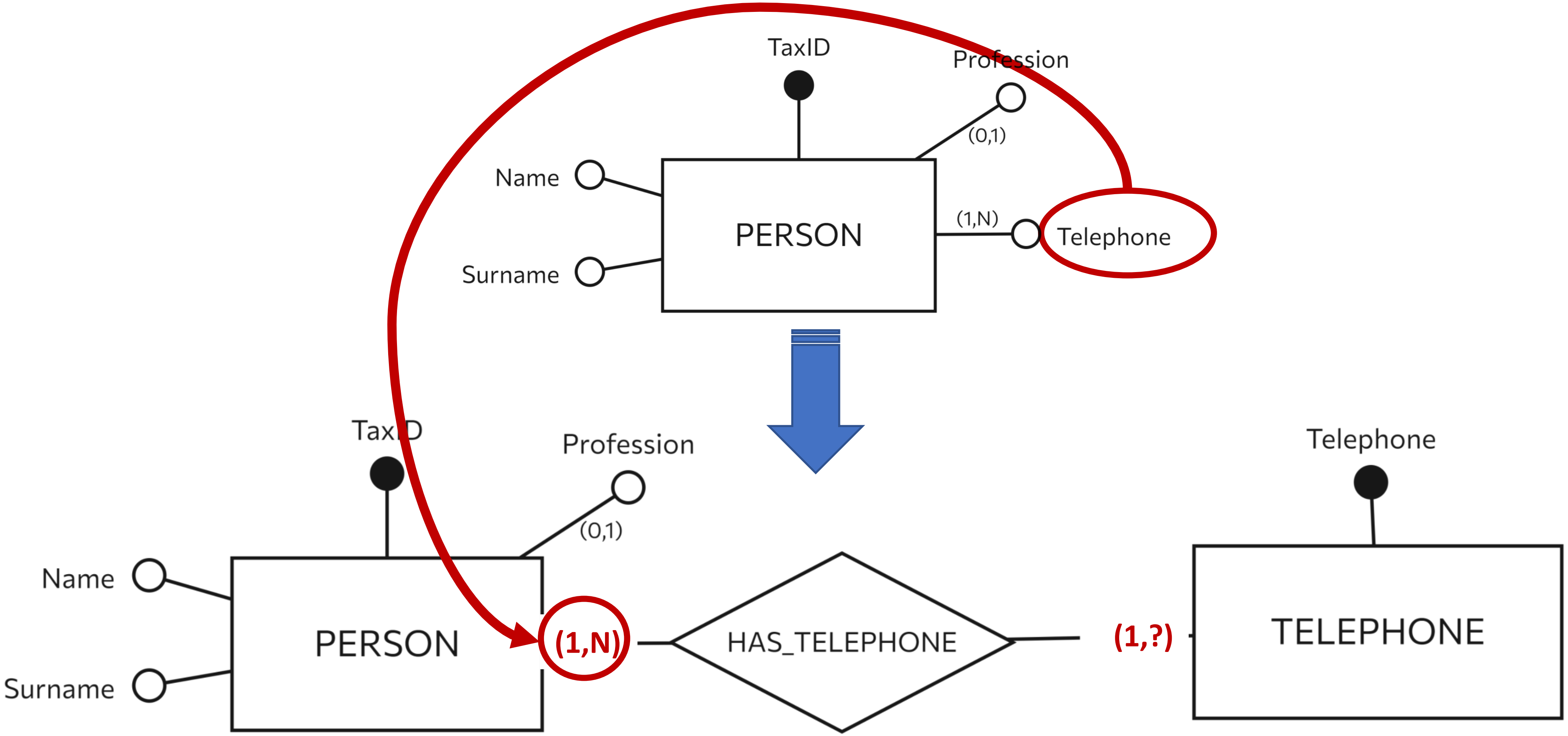
A person can have more than one educational qualification and that the same educational qualification can be held by several people

Unique information

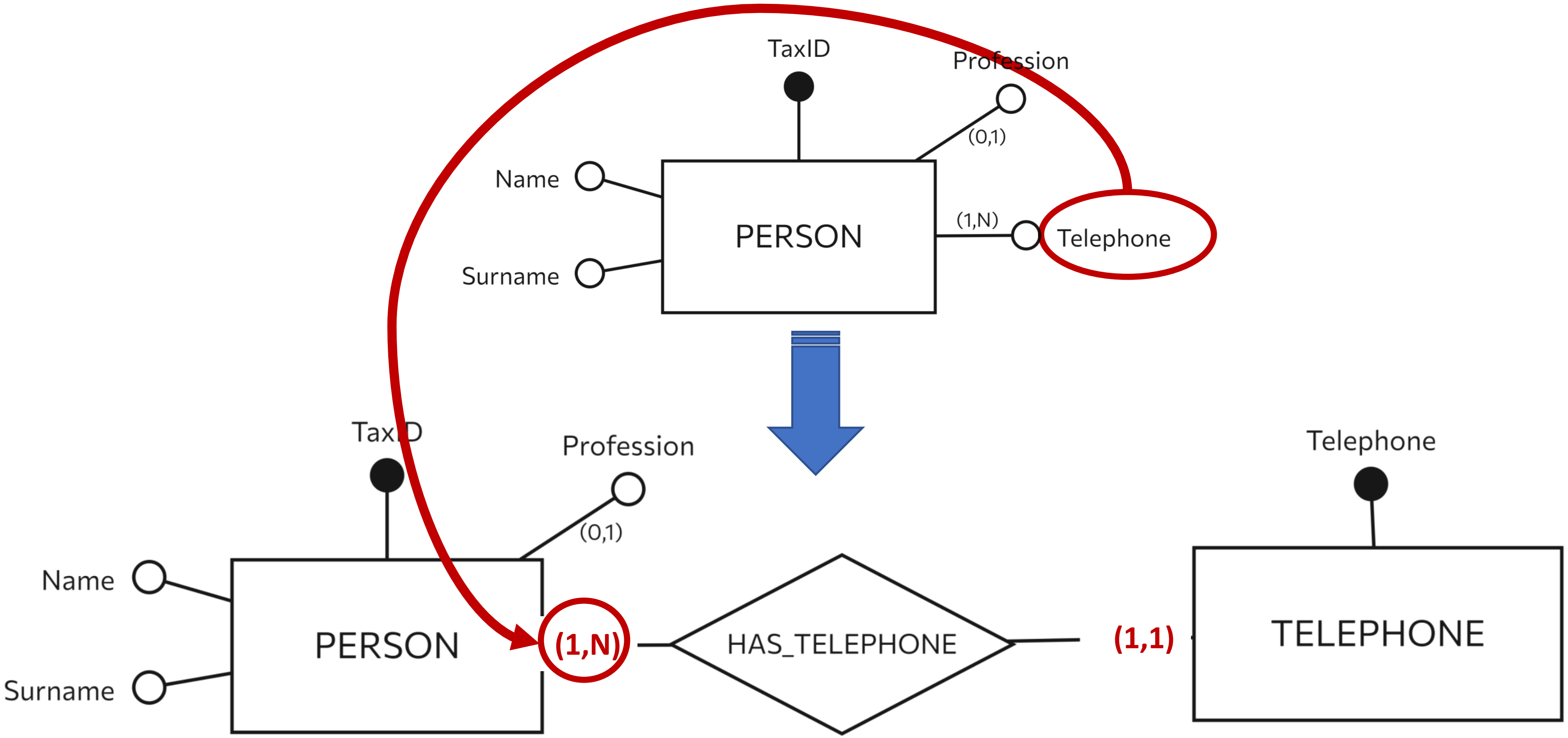


A person can have more than one telephone number, but a given telephone number can be held only by one person

Unique information: *Has_telephone* cardinality



Unique information: *Has_telephone* cardinality



A person can have more than one telephone number, but a given telephone number can be held only by one person

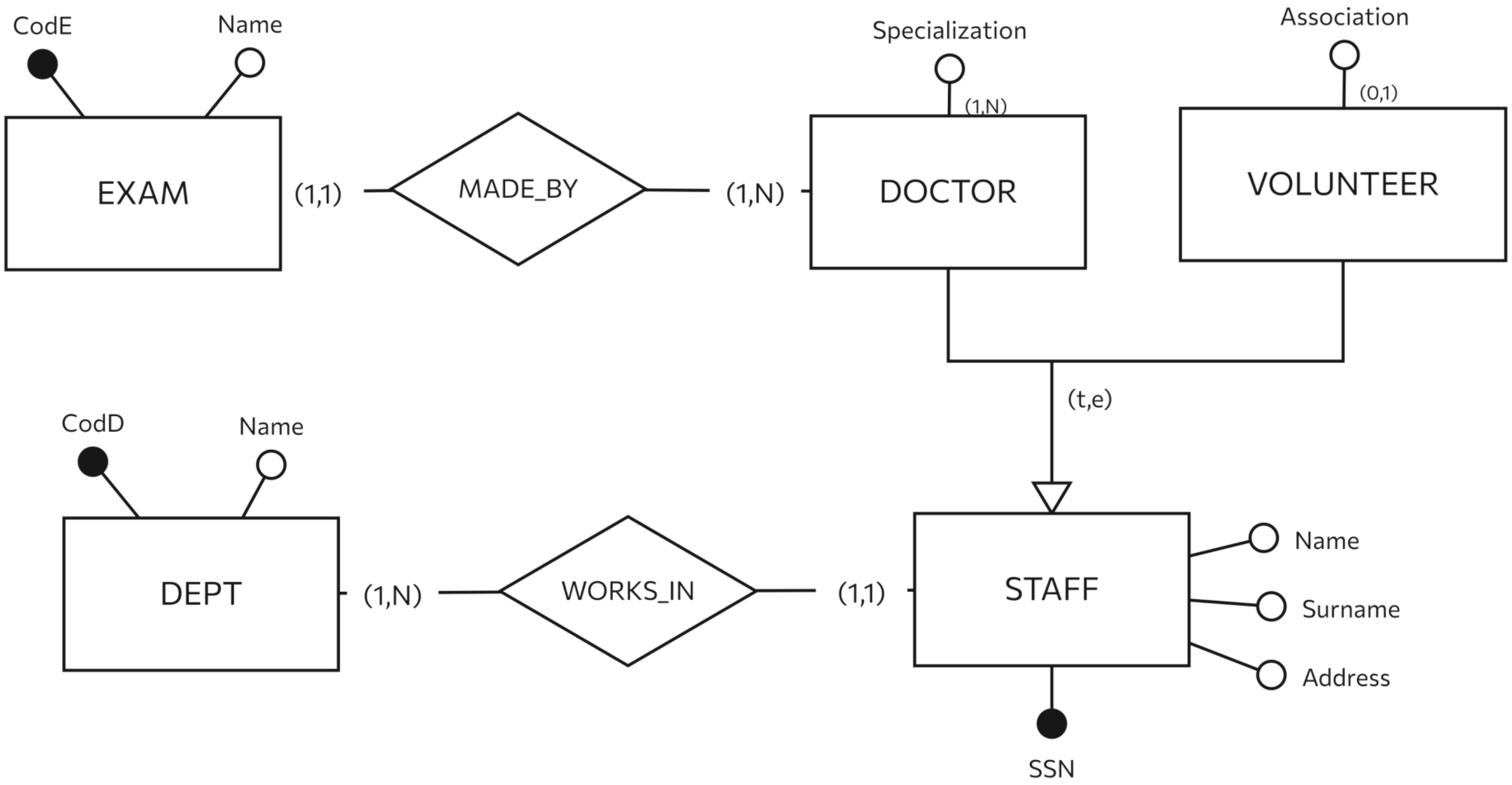
Removing generalizations

Restructuring the ER model

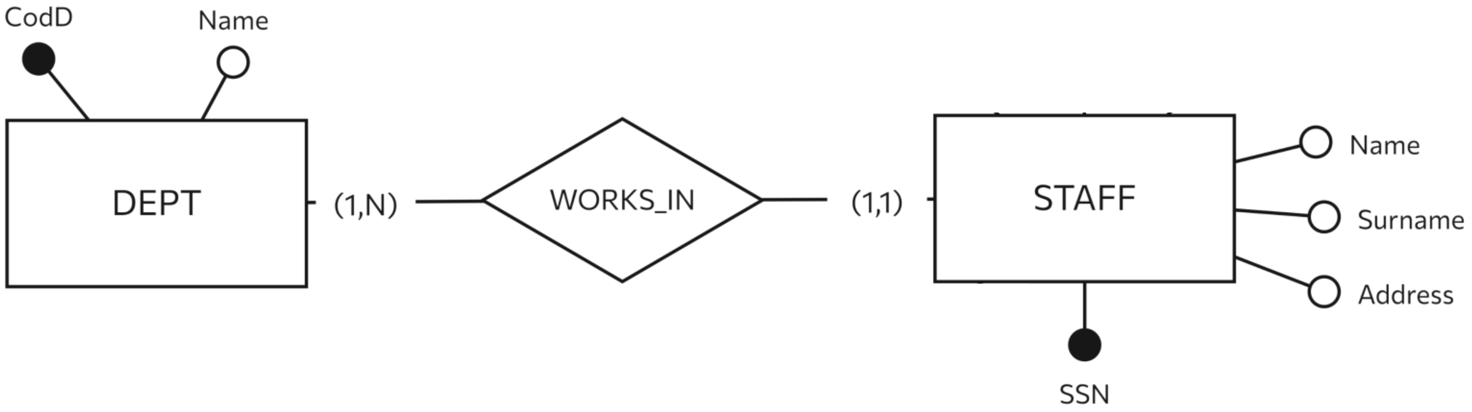
Removing generalizations

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

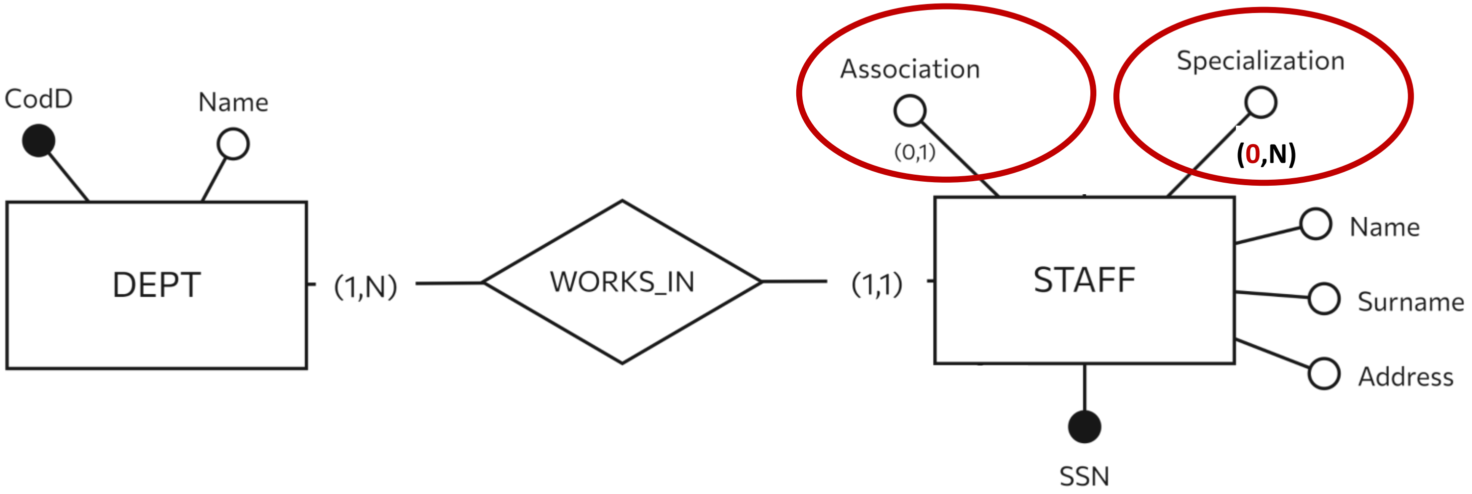
Example


CodE
Name
EXAM
(1,1)
MADE_BY
(1,N)
DOCTOR
Specialization
(1,N)
Association
(0,1)
VOLUNTEER
(t,e)
CodD
Name
DEPT
(1,N)
WORKS_IN
(1,1)
STAFF
Name
Surname
Address
SSN

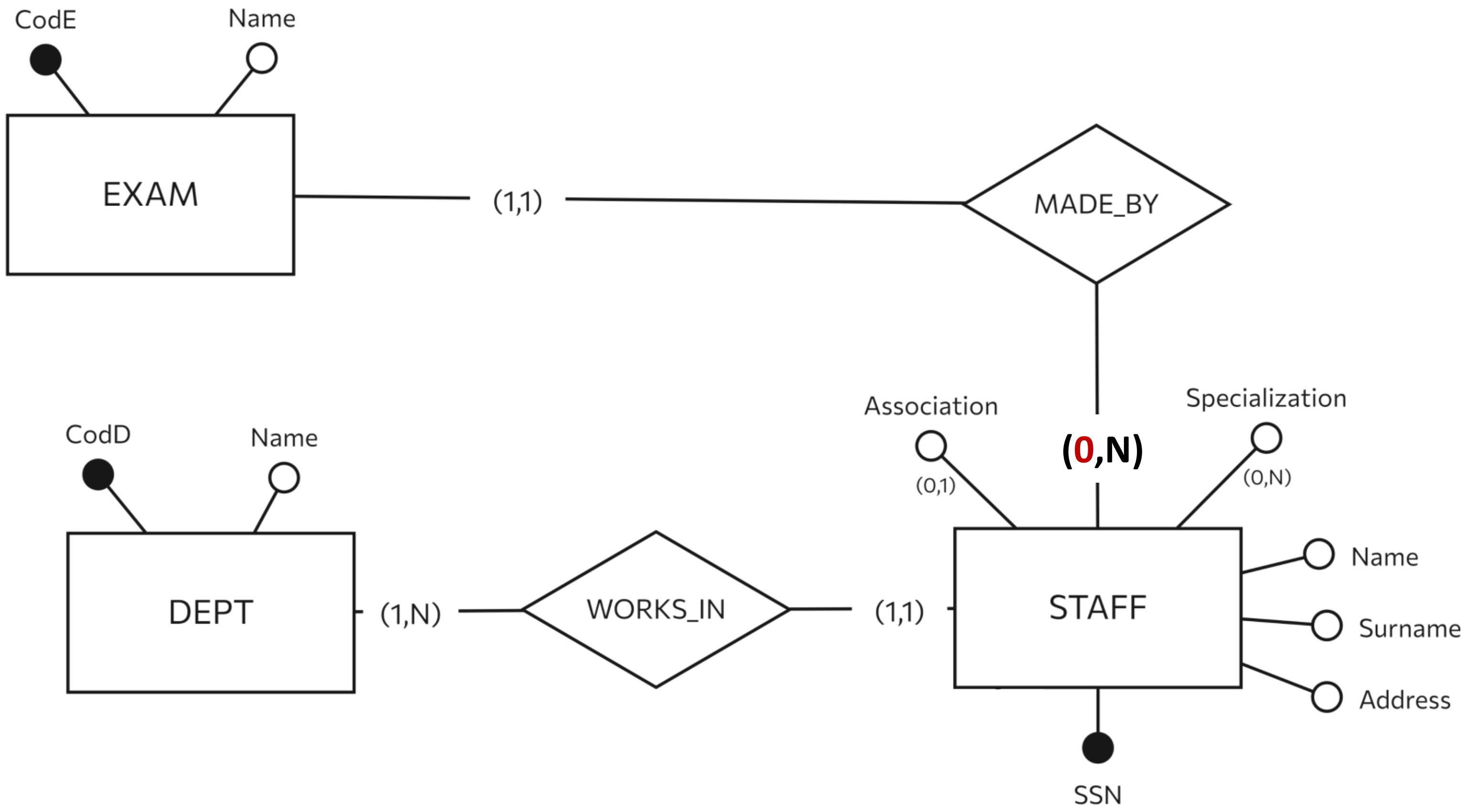
Merging child entities into the parent entity



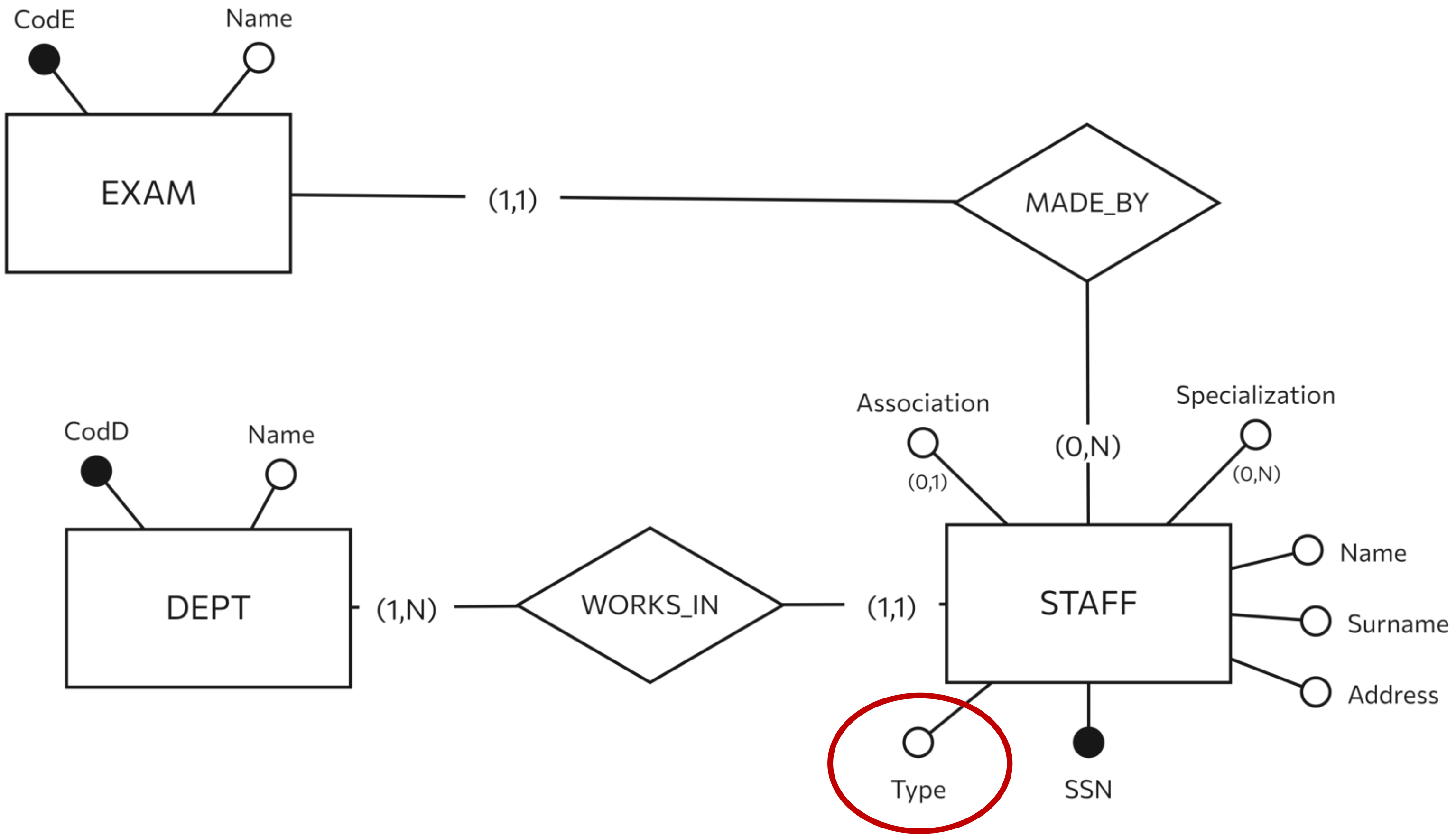
Attributes of child entities



Relationships with child entities

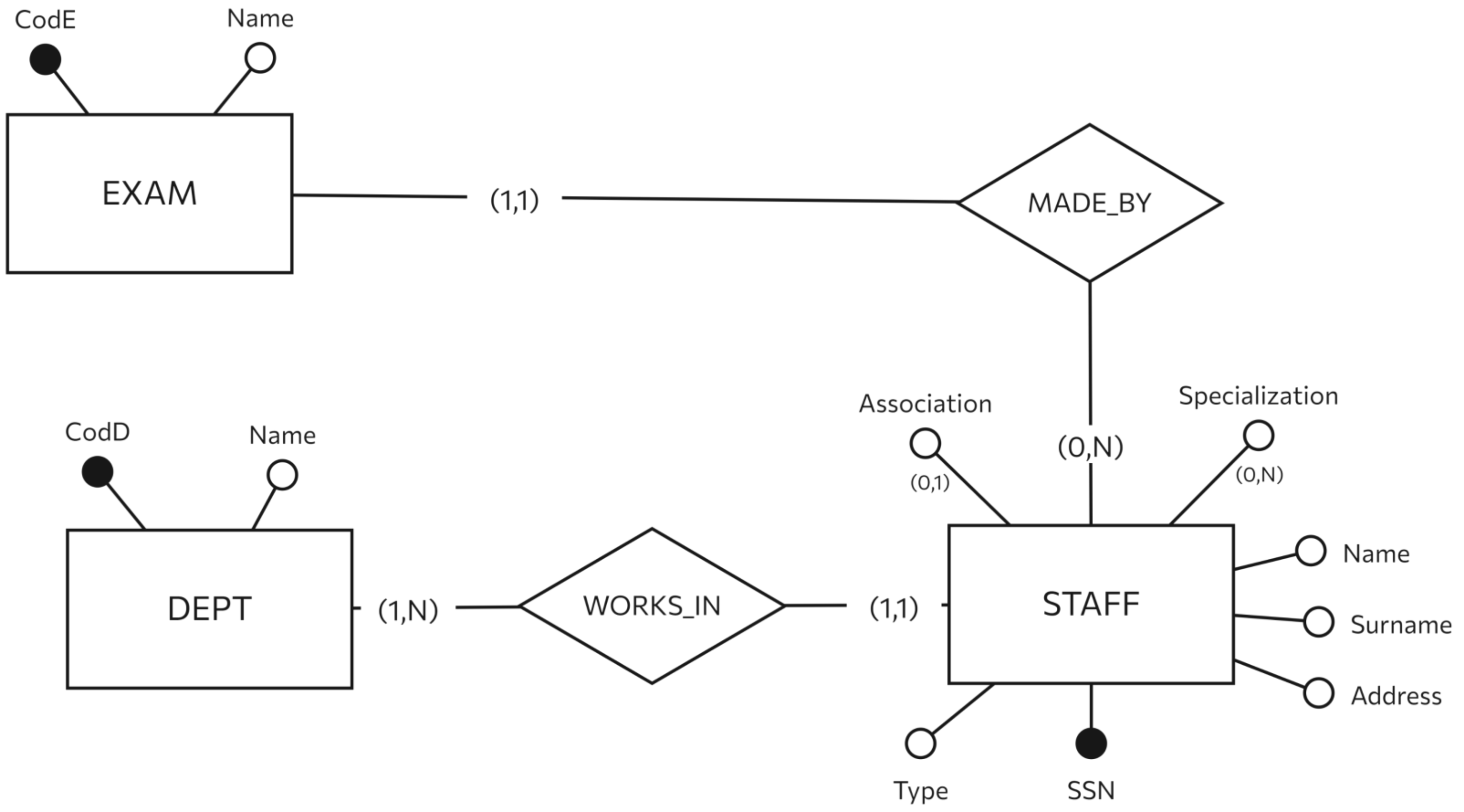


The «Type» attribute



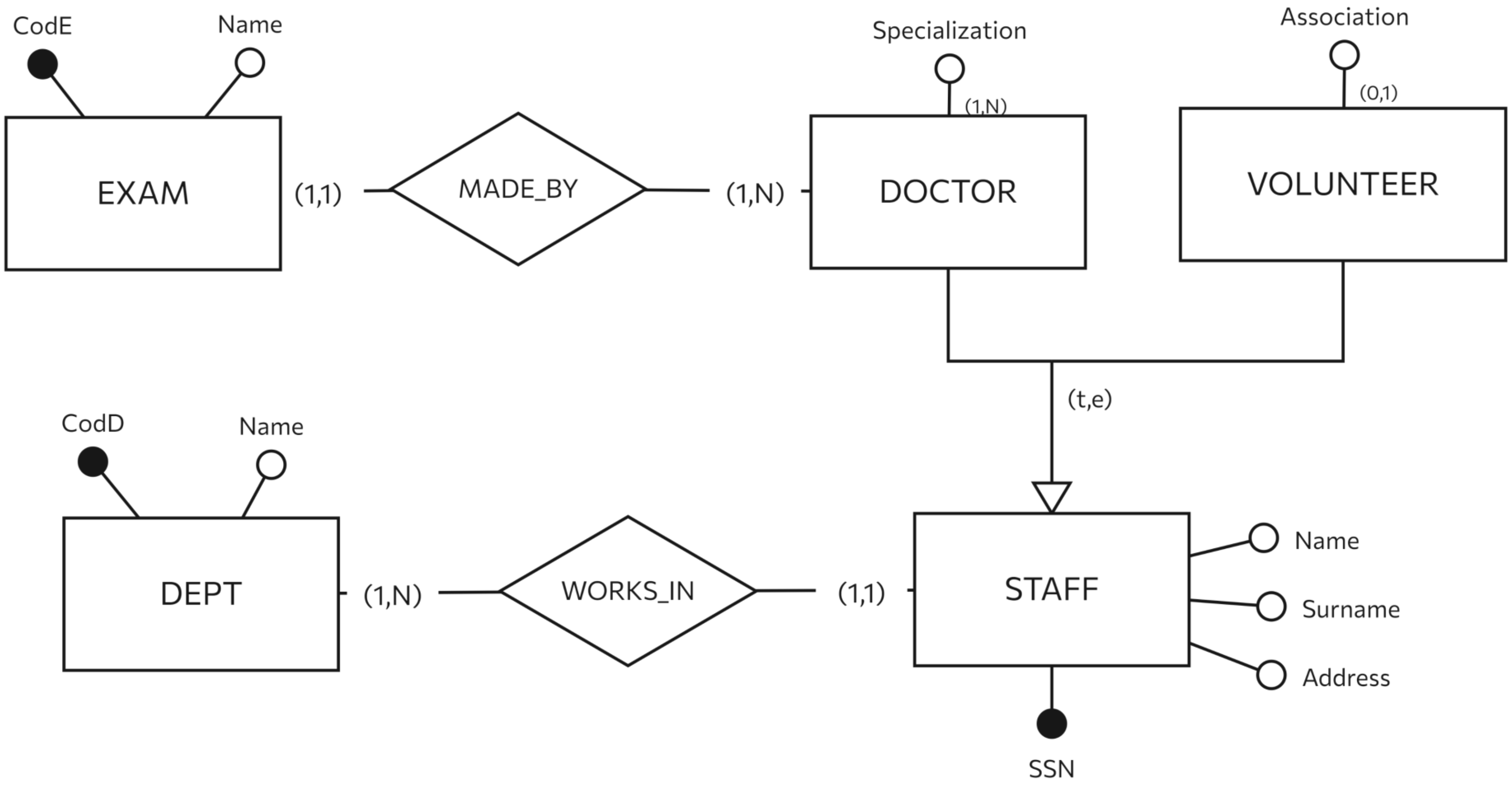
- The Type attribute indicates the original entity (doctor or volunteer) to which each occurrence of the parent entity (staff) belongs

Merging child entities into the parent entity

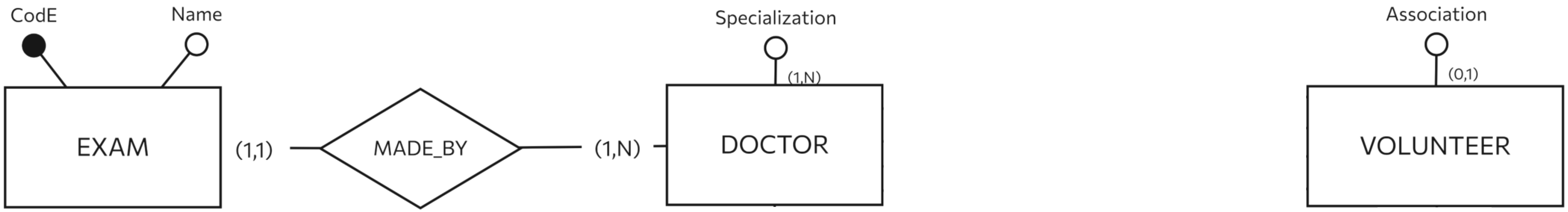


- Can be used for all types of generalization
  - in case of overlapping entities, many combinations are possible as Type values, e.g., skier and sailor

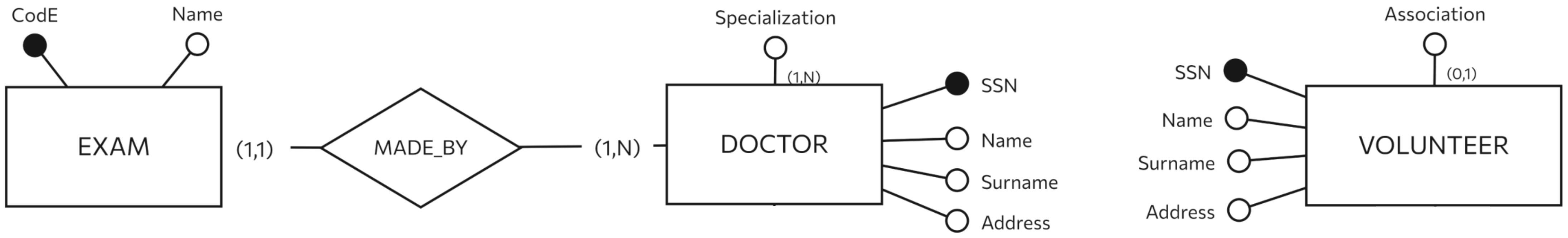
Back to the example



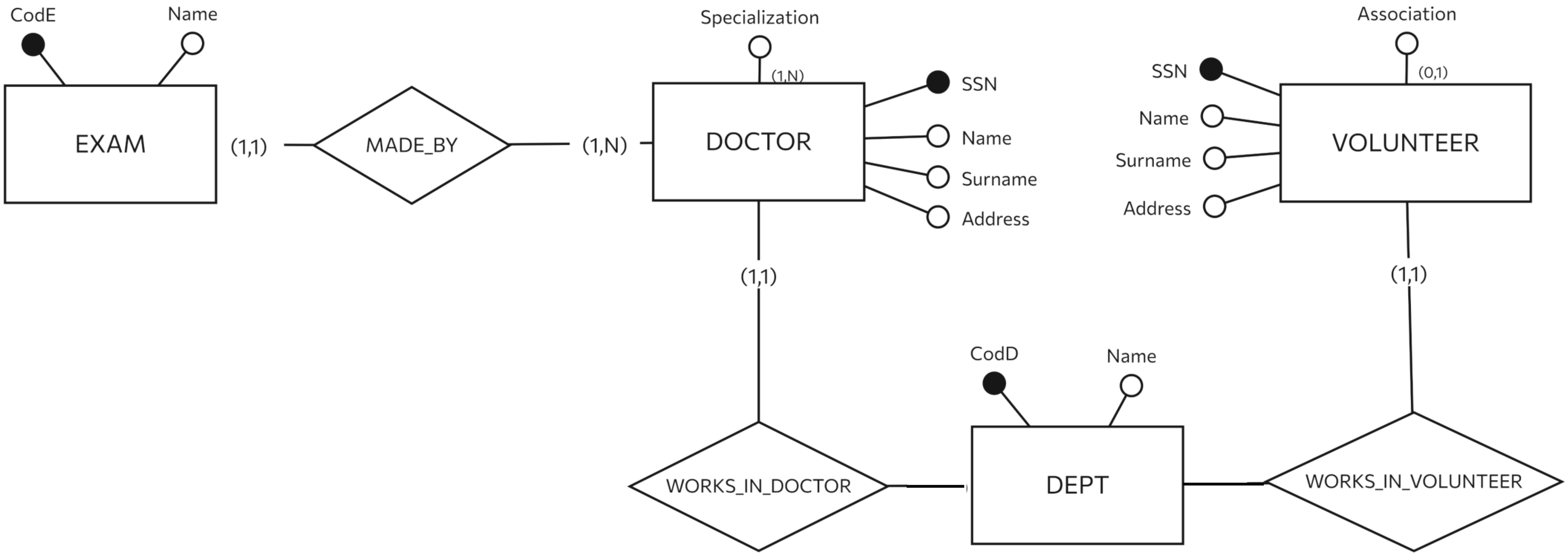
Merging the parent into the child entities



Attributes of the parent entity

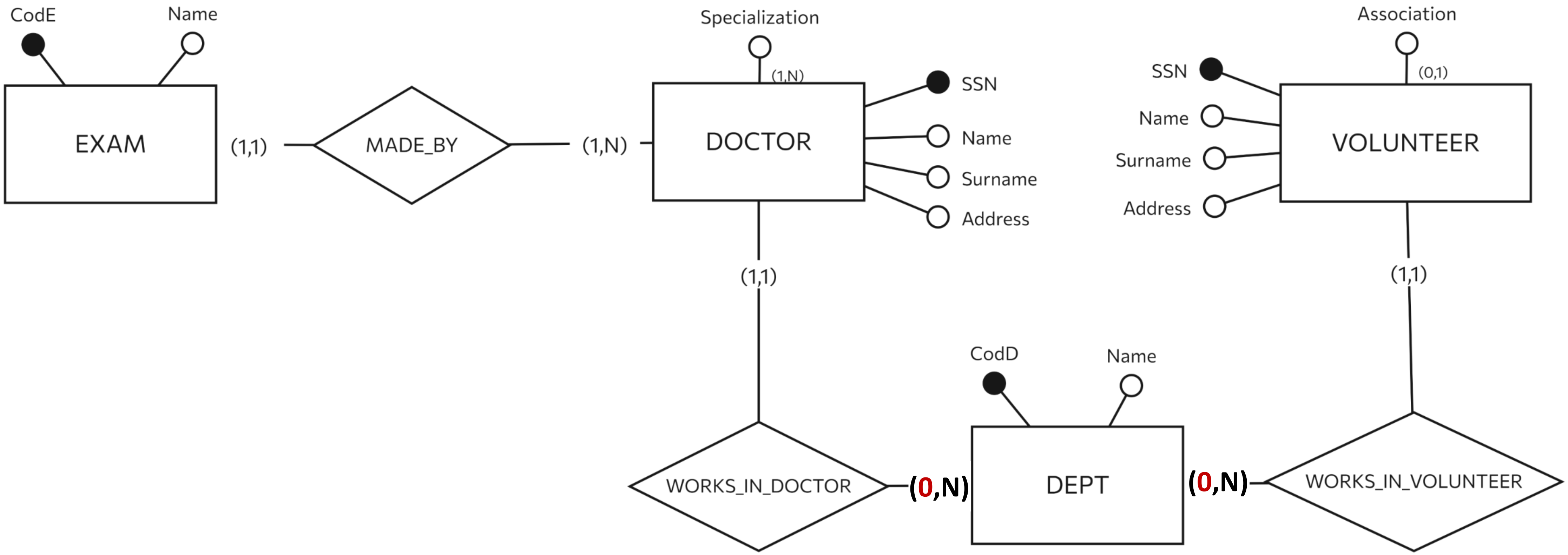


Relationships with parent entity

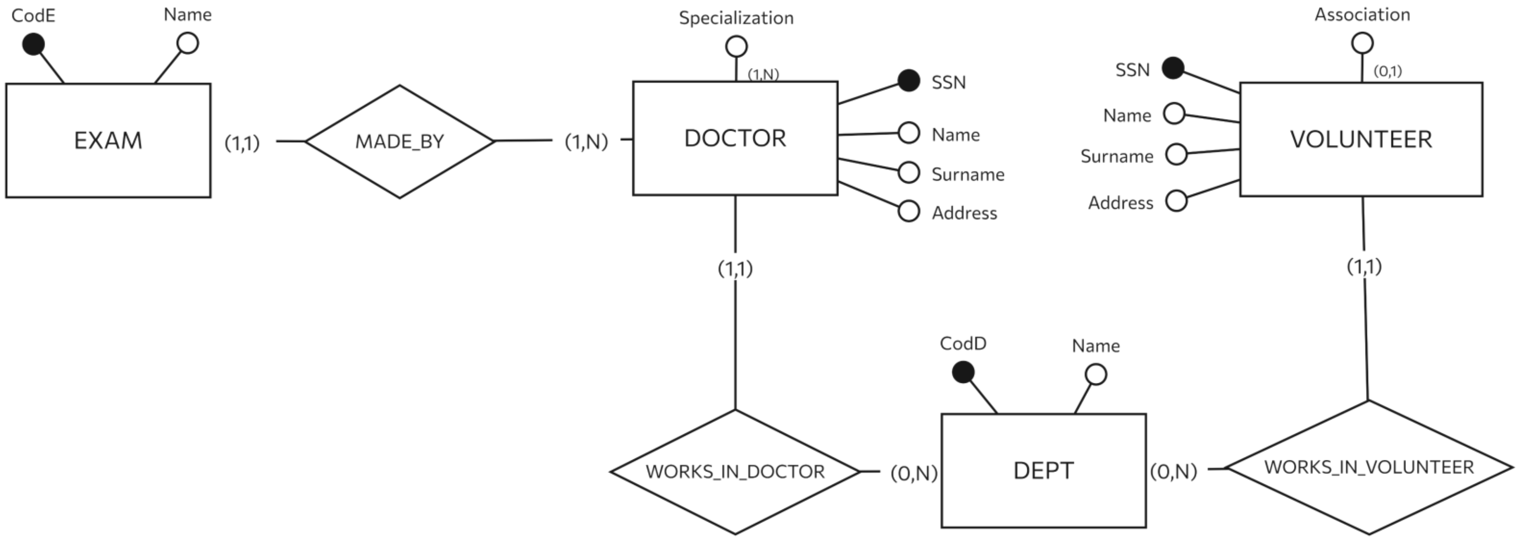


- Relationships with the parent entity need to be split

Cardinality of *Works in* relationship

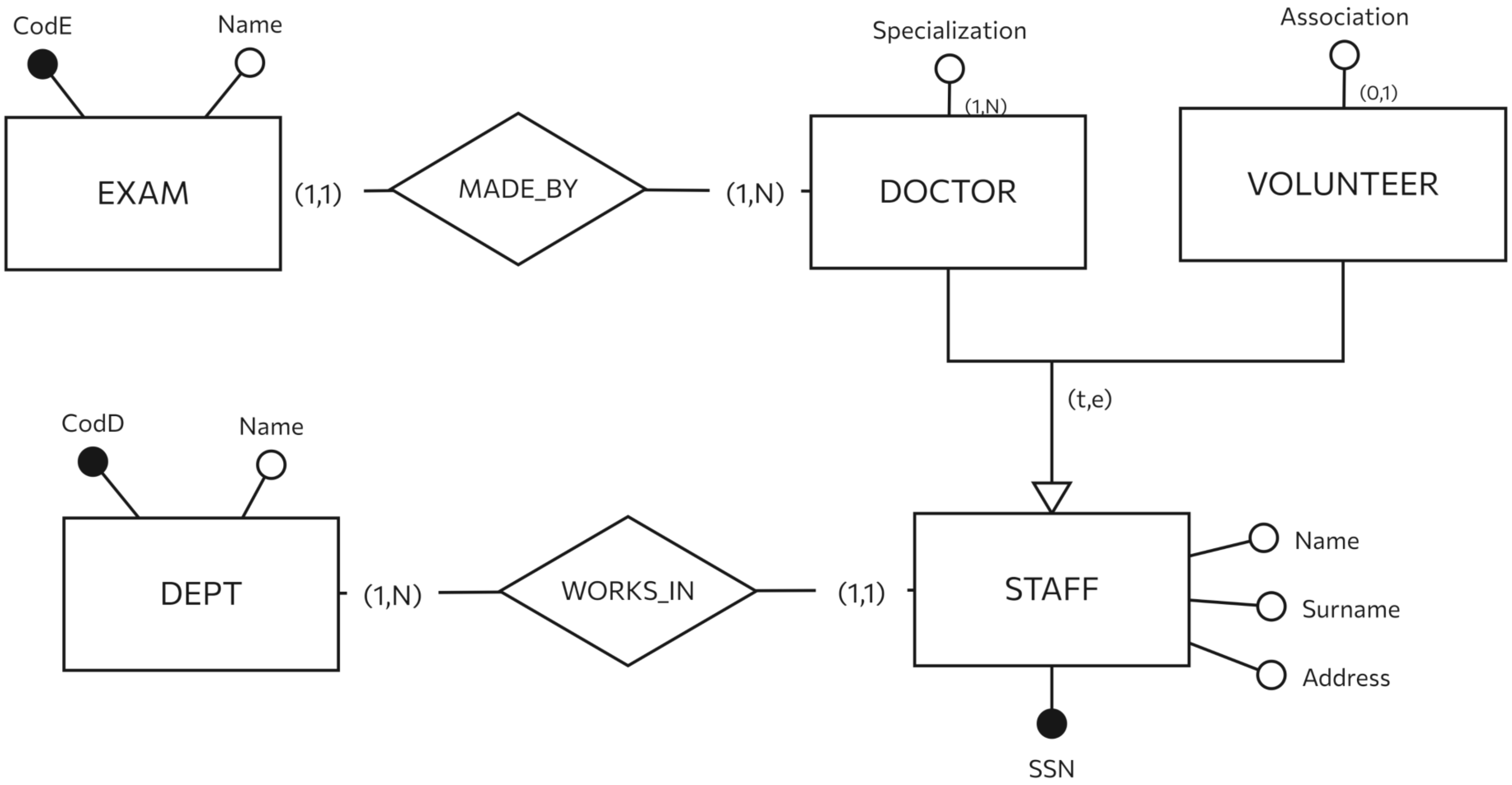


Merging the parent into the child entities

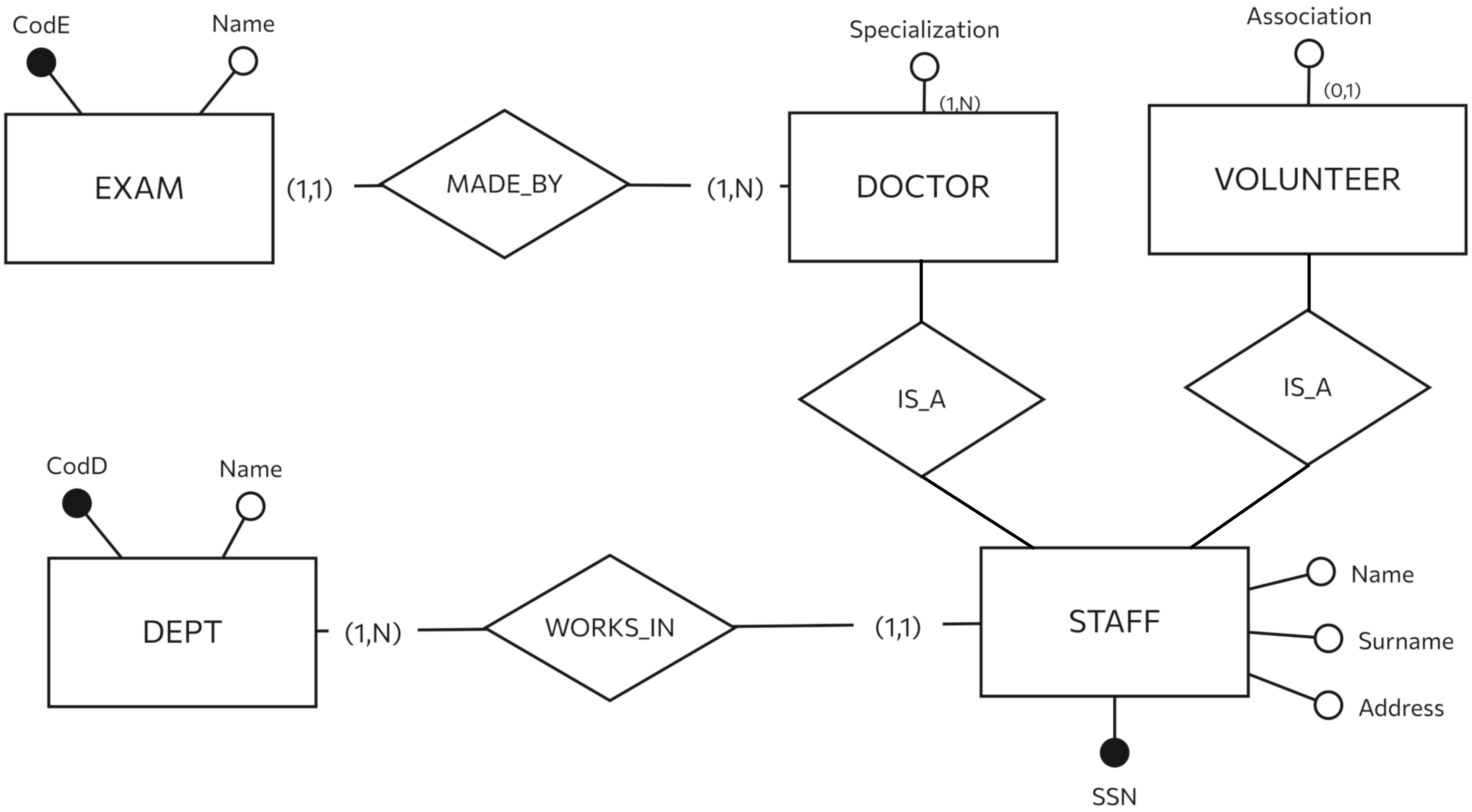


- Cannot be used for **partial** generalizations
  - however, generalizations can be transformed from partial to total by adding a new entity Others
- Cannot be used for **overlapping** generalizations
  - due to duplicate identifiers

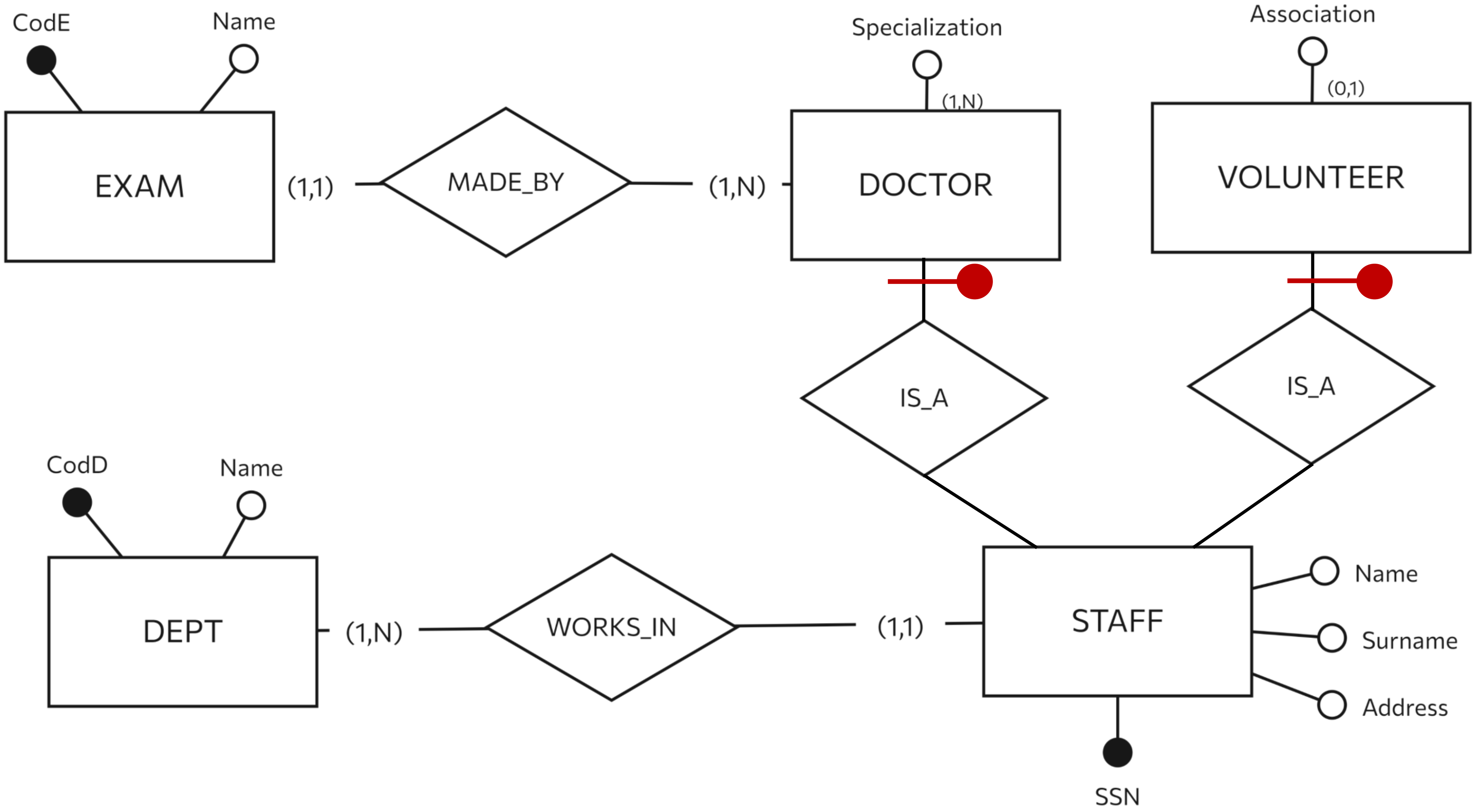
Back to the original example



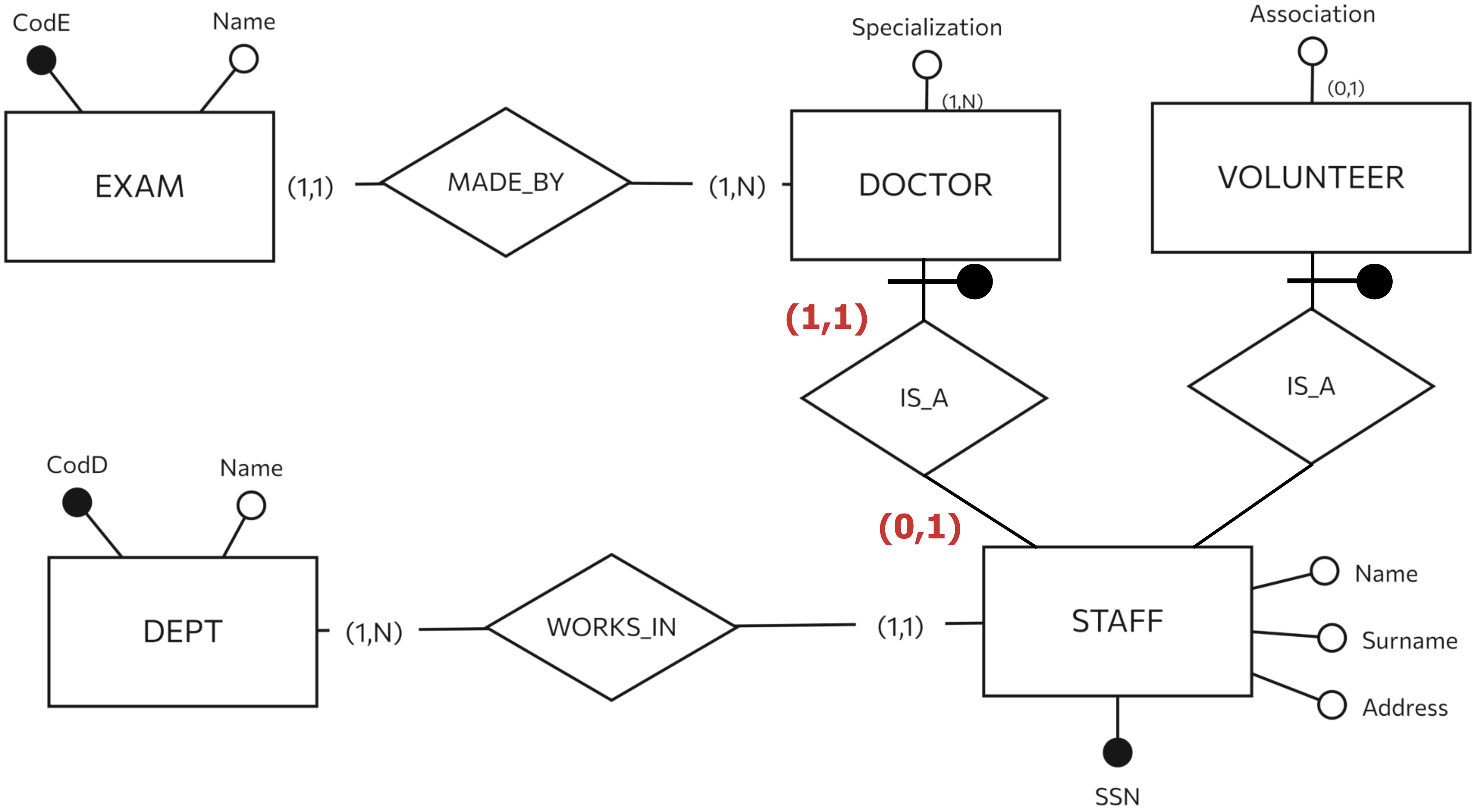
Generalization translated into a relationship



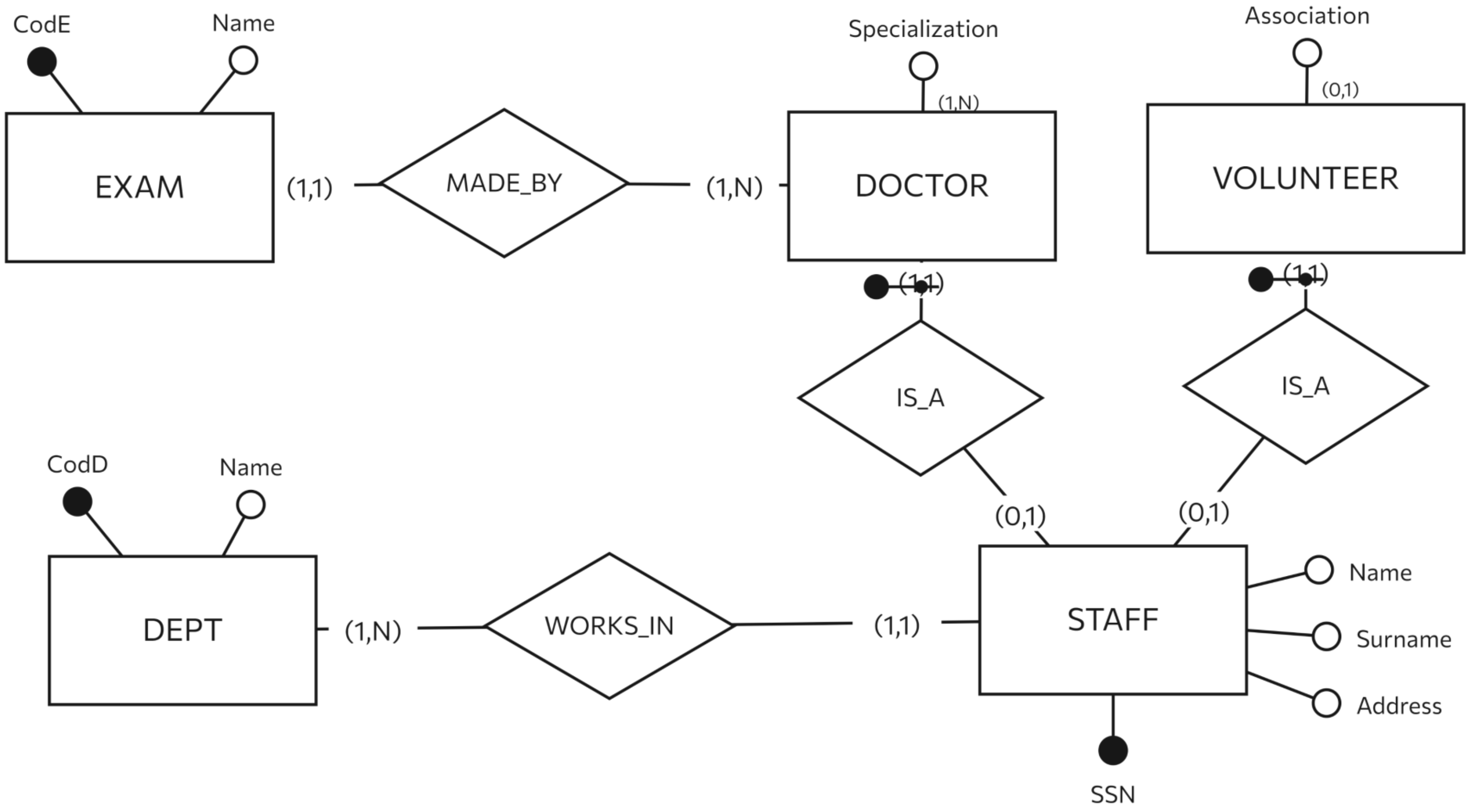
Child entities' identifier



Cardinality of *is a* relationship



Generalization translated into relationships



- This solution is more general and can be used for all generalizations
  - But it may be expensive to reconstruct the original data

Assessment of alternatives

- Merging child entities into parent entity is appropriate when:
  - access operations apply to instances and attributes of child and parent entitites more or less in the same way (optimize data access)
  - child entities are mildly differentiated (few null values)
- Merging parent entity into child entities is appropriate when:
  - the generalization is total
  - there are operations that refer only to specific child entities and therefore it is useful to distinguish between different child entities (optimize data access)
- “Mixed” representations are also possible:
  - there are operations that refer only to instances of some child entities (optimize data access)
- In the presence of hierarchical generalization, apply the same procedure, starting from the lower levels

Redundancy analysis

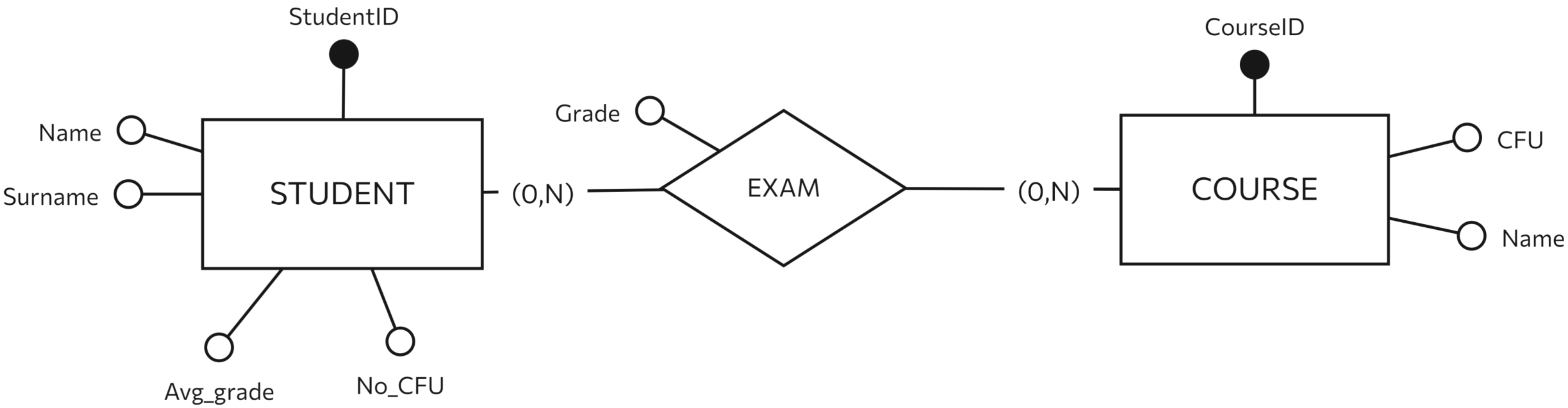
Restructuring the ER model

Redundancy analysis

- They represent information that is relevant to the application, but can be derived from other concepts
    - it must be decided whether to keep them
- Effects of redundancies on the logical schema
    - simplifying and speeding up queries
    - increased complexity and slower updates
    - increased storage requirements

Redundant attribute example

- The Avg_grade attribute is redundant:
  - Useful for speeding up queries that require calculating the average of students' grades
  - if preserved, the relational schema must be supplemented with proper documentation that the attribute is redundant (and derivation rules)



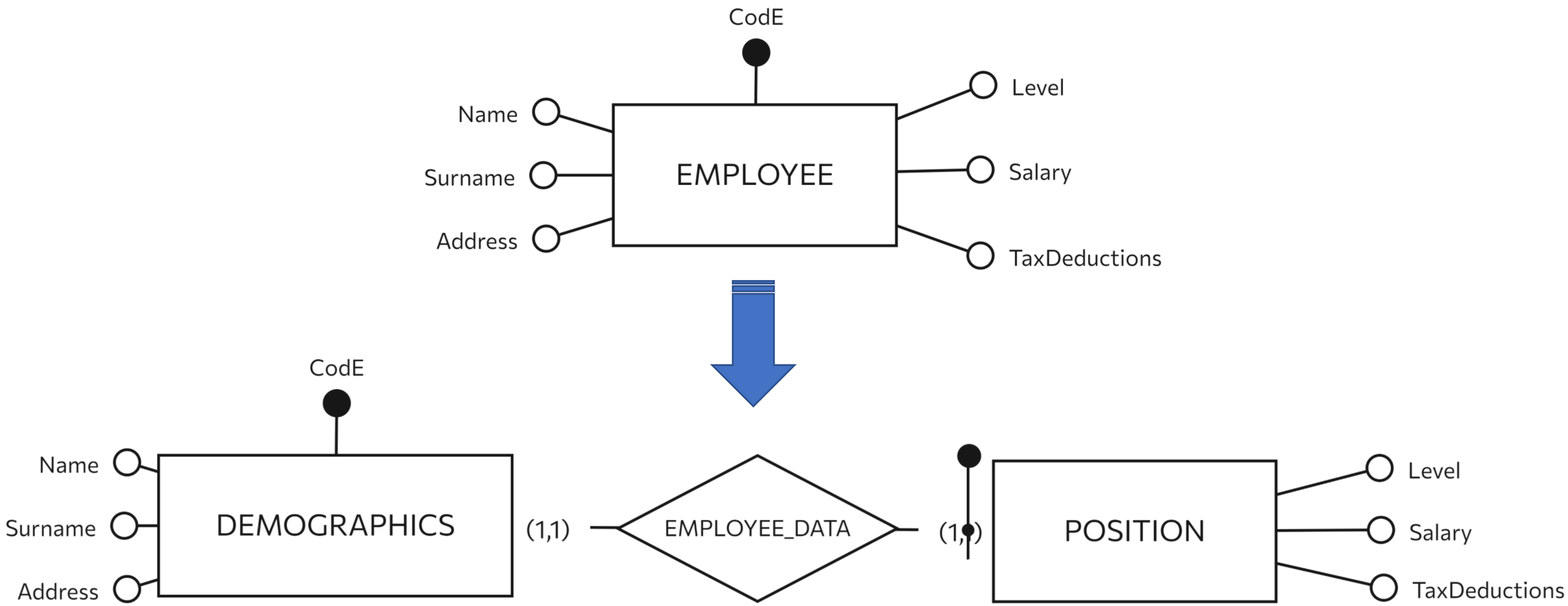
Partitioning concepts

Restructuring the ER model

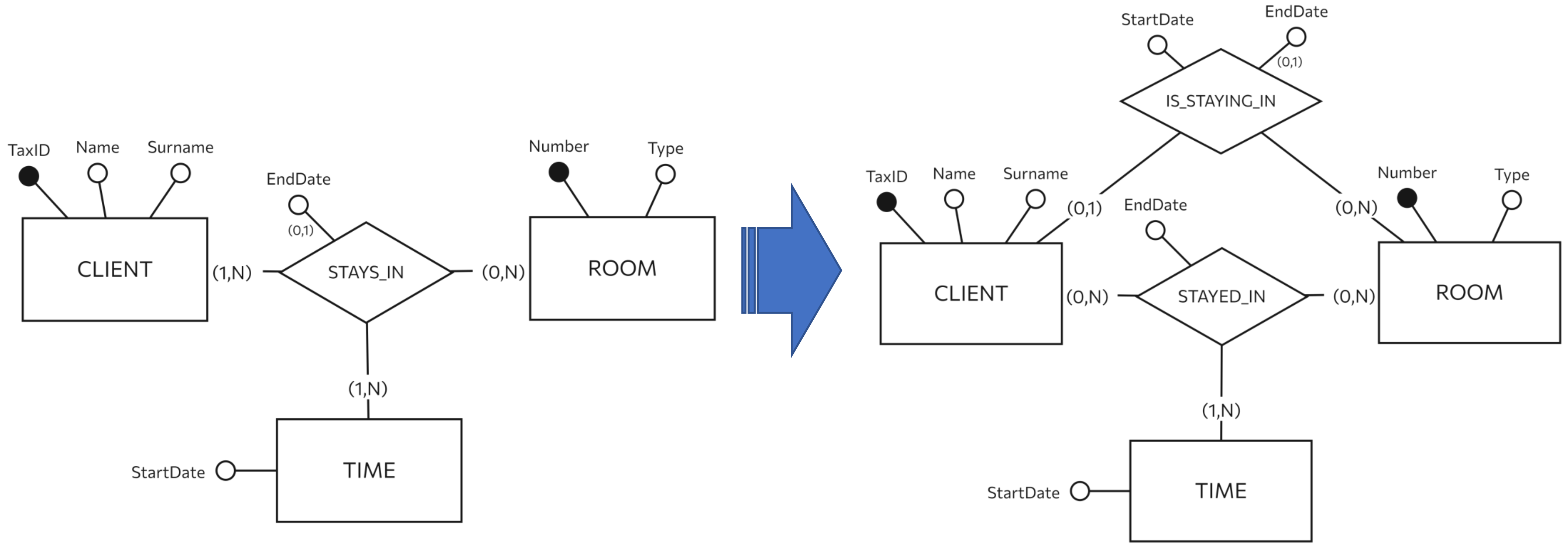
Partitioning of concepts

- Partitioning of entities and relationships
  - better representation of different concepts
  - separating attributes of the same concept that are accessed by different operation
  - improve the efficency of the operations

Entity partitioning



Relationship partitioning



Choosing Primary Identifiers

Restructuring the ER model

Selection of primary identifiers

- It is necessary to define the relation *primary keys*
- The criteria for this decision are as follows
  - Attributes with **null** values **cannot** form primary identifiers.
  - Just **one** (better) or **few** attributes
  - An **internal** identifier is preferable to an external one
  - It is used by many operations to access the occurrences
- It may be useful to introduce an additional attribute to represent the entity, often called code or ID, e.g. «ProductCode»