Conceptual Database Design with ER Models

Alessandro Bozzon
TI1506: Web and Database Technology
ti1506-ewi@tudelft.nl

TU Delft

Course overview [DB]

1. Introduction to Database Systems
2. The Relational Model
3. Introduction to SQL
4. Advanced SQL Topics
5. Introduction to NoSQL database systems
6. **Conceptual Data Modelling with ER Diagrams**
7. Database Conceptual Design
8. Database Logical Design

At the end of this lecture, you should be able to …

- **Develop** and **refine** conceptual data models, using appropriate design notation

Introduction To Database Design

Database Design

- Information Systems design:
  - **Database**
  - Applications
  - User interfaces
  - Other service programs

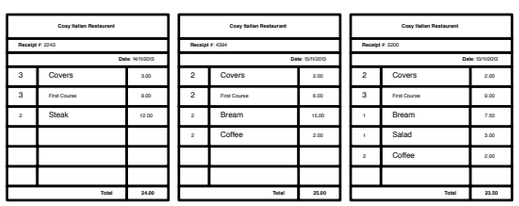
- Data plays a **central role** in an information system

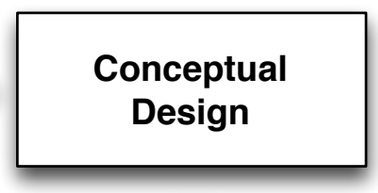
- It justifies an independent study of database (**data**) design
  - requirements collection and analysis

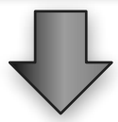
A Methodology for Database Design

- Simple but highly efficient engineering principle

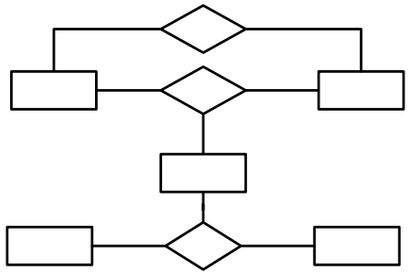
Split the decisions about
what to represent in the database (conceptual design) and **how** to represent it (logical and physical design)

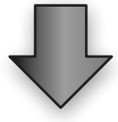


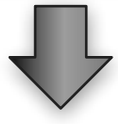




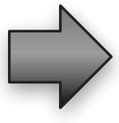
Conceptual Schema

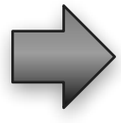




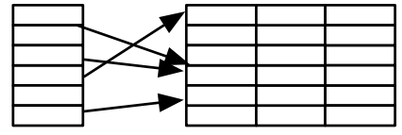


Logical Schema





Physical Schema



A subset of steps in Section 7.1

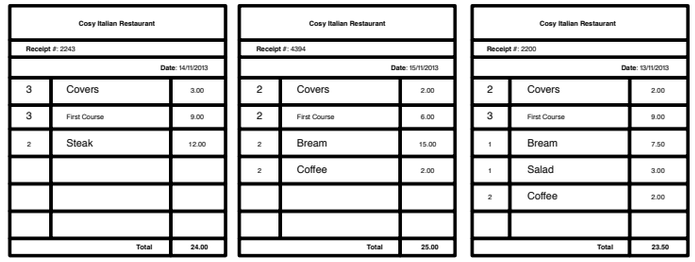
Receipts		
Number	Date	Total
2200	13/11/2013	23.50
2243	14/11/2013	24
4394	14/11/2013	25

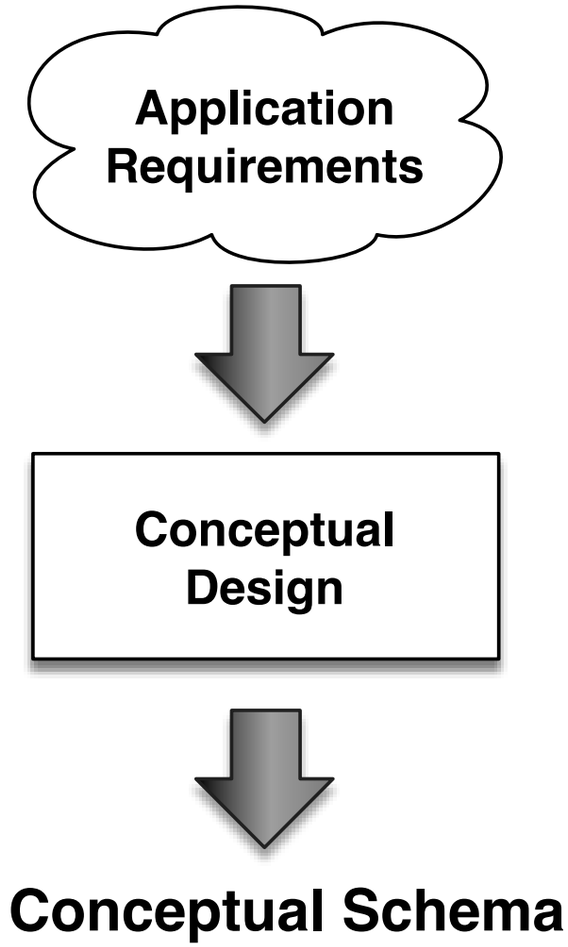
Details			
Number	Quantity	Item	Line
2220	2	I001	1
2220	3	I002	2
2220	1	I003	3
2220	1	I004	4
2220	2	I006	5
2243	3	I001	1
2243	3	I002	2
2243	2	I005	3
4394	2	I001	1
4394	2	I002	2
4394	2	I003	3
4394	2	I006	4

Item		
Number	Description	Cost
I001	Covers	1.00
I002	First Course	3.00
I003	Bream	2.50
I004	Salad	1.00
I005	Steak	6.00
I006	Coffee	1.00

I) Conceptual Design

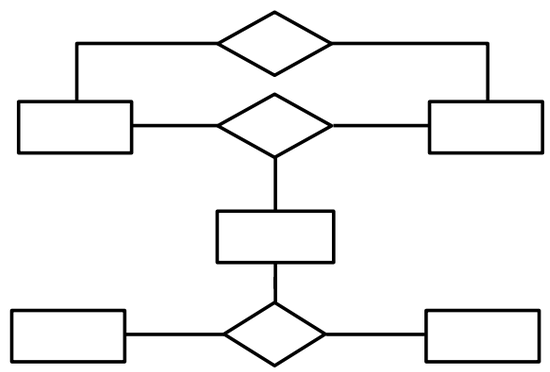
Represent the informal requirements of an application in terms of a conceptual schema that refers to a conceptual data model





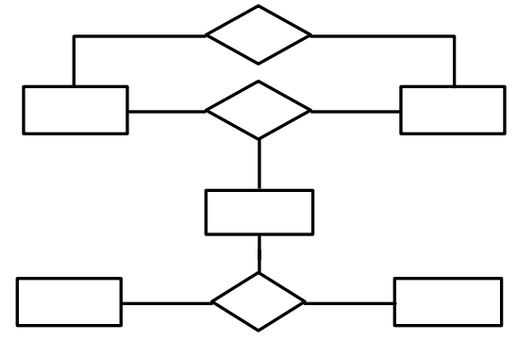
- **GOAL:** represent the informal requirements of an application in terms of a formal and complete description
  - Independent from the DBMS of choice
  - (Mostly) Independent from performance considerations

- **OUTPUT**: a conceptual schema, referring to a conceptual data model

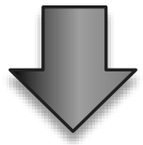


2) Logical Design

Translation from conceptual schema to the logical schema of the database that refers to a logical data model



Conceptual Schema



Logical Design

Logical Schema

- **GOAL:** represent the conceptual schema using the data model used by the DBMS of choice
  - Independent from physical details
  - Optimisation according to the operations to be performed on the database

- **OUTPUT**: a logical schema, referring to a logical data model (e.g. relational)
  - Quality of the resulting schema can be formally verified (*normalisation*)

Receipts		
Number	Date	Total
2200	13/11/2013	23.50
2243	14/11/2013	24
4394	14/11/2013	25

Details			
Number	Quantity	Item	Line
2220	2	I001	1
2220	3	I002	2
2220	1	I003	3
2220	1	I004	4
2220	2	I006	5
2243	3	I001	1
2243	3	I002	2
2243	2	I005	3
4394	2	I001	1
4394	2	I002	2
4394	2	I003	3
4394	2	I006	4

Item		
Number	Description	Cost
I001	Covers	1.00
I002	First Course	3.00
I003	Bream	2.50
I004	Salad	1.00
I005	Steak	6.00
I006	Coffee	1.00

3) Physical Design

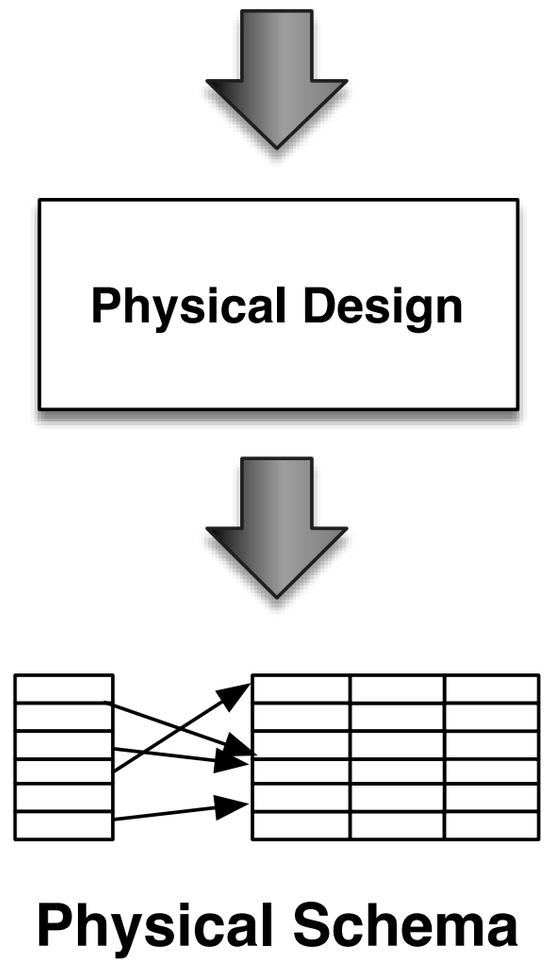
Specialisation of the logical schema into a physical schema and refers to a physical data model

Details			
Number	**Quantity**	**Item**	**Line**
2220	2	I001	1
2220	3	I002	2
2220	1	I003	3
2220	1	I004	4
2220	2	I006	5
2243	3	I001	1
2243	3	I002	2
2243	2	I005	3
4394	2	I001	1
4394	2	I002	2
4394	2	I003	3
4394	2	I006	4

Item		
Number	**Description**	**Cost**
I001	Covers	1.00
I002	First Course	3.00
I003	Bream	2.50
I004	Salad	1.00
I005	Steak	6.00
I006	Coffee	1.00

Logical Schema

- **GOAL:** complete the logical schema with details of the physical implementation
  - E.g. file organisation and indexes on a given DBMS
  - The physical data model depends on the adopted DBMS
- **OUTPUT**: a physical schema, referring to a physical data model



Physical Schema

Data and Operational Requirements

- **Requirements collection**
  - precedes (or overlaps) with **Database design**
  - GOAL: **Identify** 1) problems that the information system must solve and 2) features of the information system

- Database design deals with at least two types of requirements
  - **data requirements**, i.e. the content of the database
  - **operational requirements**, i.e. the use of the database by users or programs

- **Conceptual design**
  - data requirements provide most of the information
  - operational requirements are used only to verify that the conceptual schema is complete

- **Logical design**
  - conceptual schema summarises data requirements
  - operational requirements together with the predicted application load are used to obtain a logical schema

- **Physical design**
  - logical schema, operational requirements, and DBMS characteristics are used to optimise performance

ASQ Exercise (45 sec)

Which of the following statements about the conceptual design step and conceptual models is correct?

1

The purpose of conceptual design is to allow the description of the organization of data at a high level of abstraction

2

The purpose of the conceptual design step is to represent the informal requirements of an application in terms of a formal and complete description.

3

In the conceptual design step, the efficiency of the under of the programs that make use of the database should be considered

4

The purpose of conceptual design is to allow the description of the organisation of data from the point of view of database users

ASQ Exercise (45 sec)

Which of the following statements about the logical design step and logical models is correct?

1. A logical model represents data in a way that is dependent of the physical (implementation) details

2. In the logical design step, the efficiency of the operations to be carried on the data should be considered

3. A logical model must consider the data model supported by the targeted database management system.

4. The purpose of logical design step is to allow the description of the physical (implementation) details of a database

The Entity Relationship Model

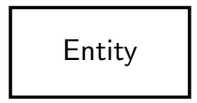
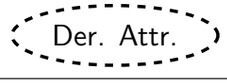
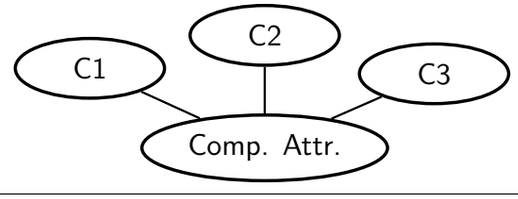
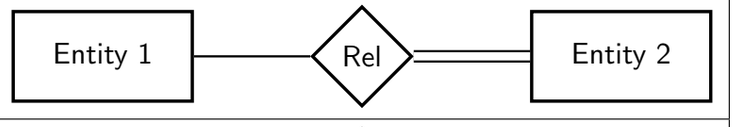
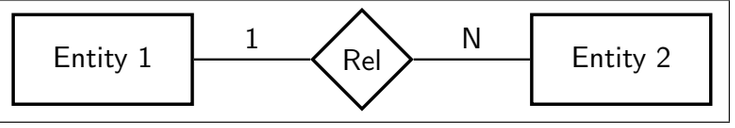
The Entity-Relationship Model

- The most used ***conceptual data model***

- Provides a series of constructs capable of *describing the data requirements* of an application:
  - in a way that is <u>easy to understand</u>
  - using a <u>graphical</u> formalisms
  - <u>independently</u> from the database system of choice

- For every construct, there is a corresponding graphical representation
  - This representation allows us to define an E-R schema diagrammatically

The constructs of the E-R model and their graphical representation

Construct	Graphical Representation
Entity	
Weak Entity	
Relationship	
Identifying Relationship	
Attribute	
Key Attribute	
Multi Valued Attribute	
Derived Attribute	
Composite Attribute	
Total Partecipation	
1:N Cardinality	

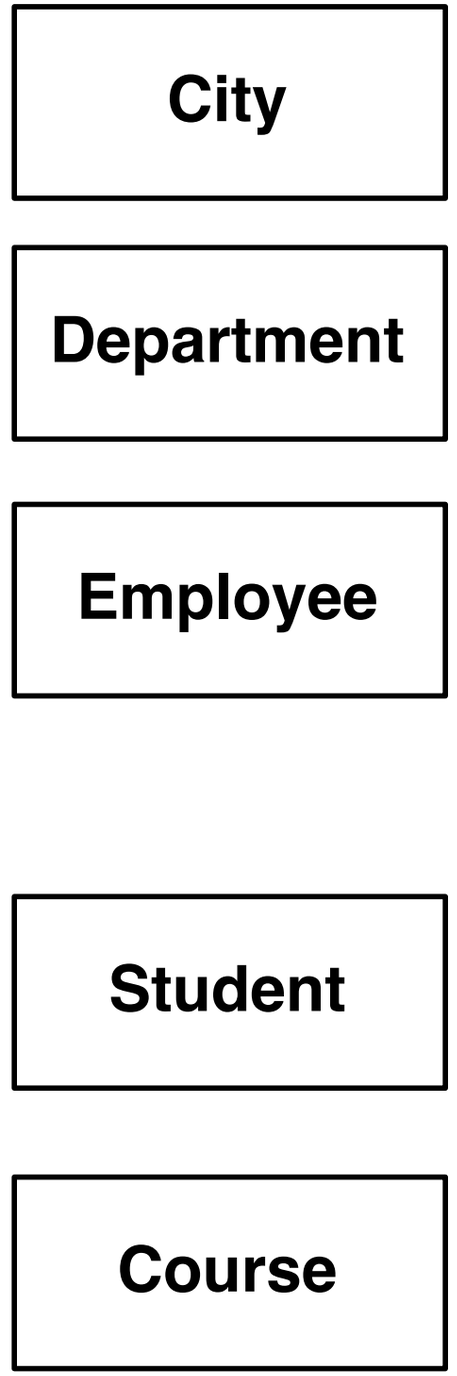
Entity and Relationship Types

Entity Types

- **Classes of objects** (e.g. facts, things, people) having:
  - common properties
  - autonomous existence

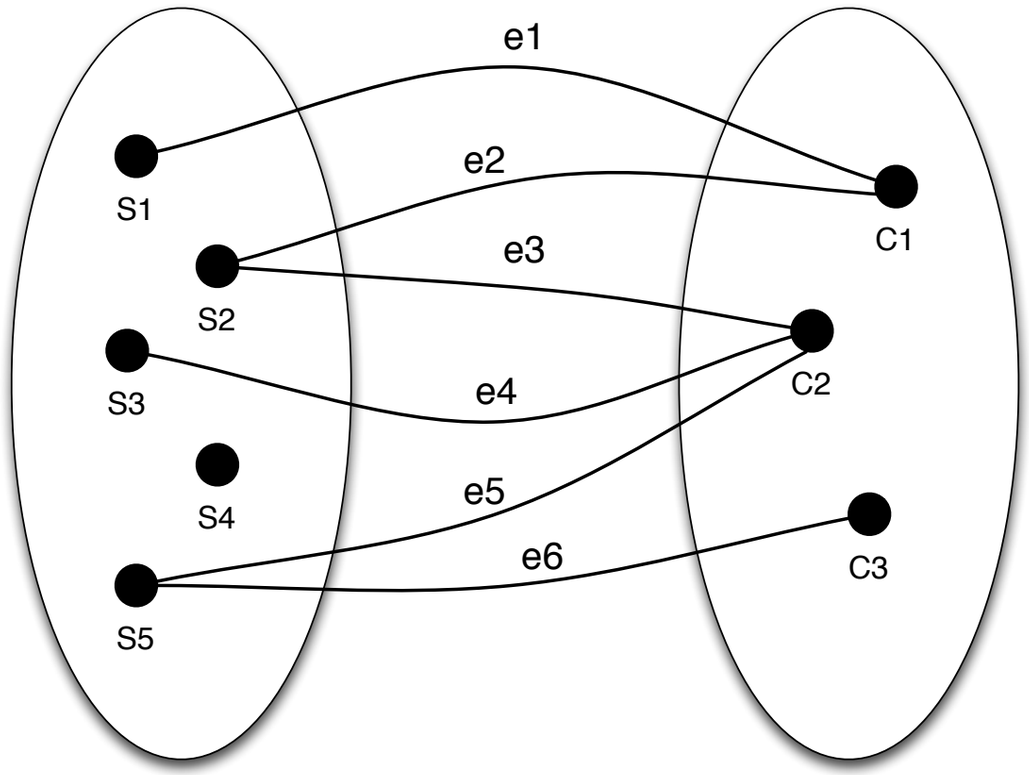
- Examples:
  - Commercial organisation: `CITY`, `DEPARTMENT`, `EMPLOYEE`, `PURCHASE` and `SALE`
  - University: `STUDENT`, `COURSE`

- An **occurrence** of an entity type is *an object* (or an *entity*) of the class that the entity type represents.



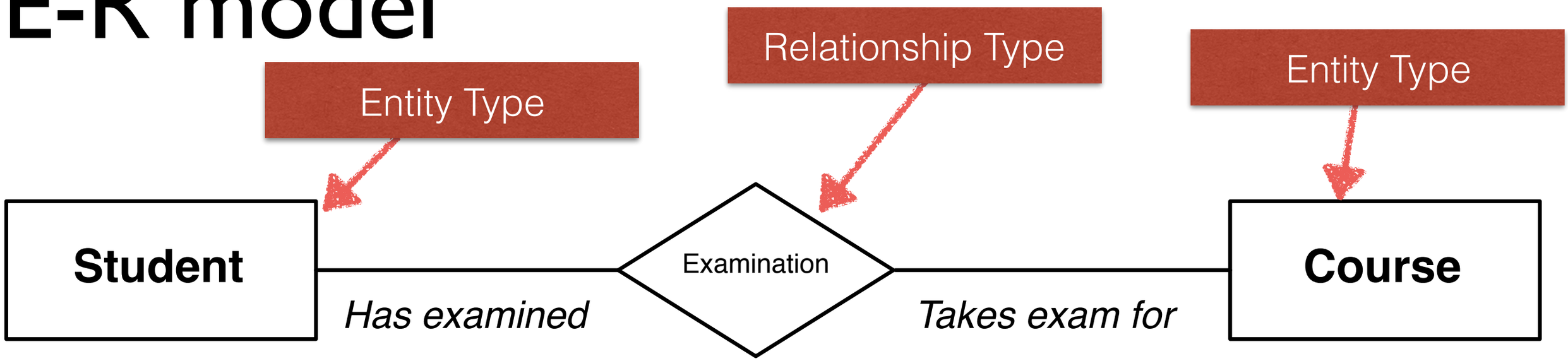
Relationship Types

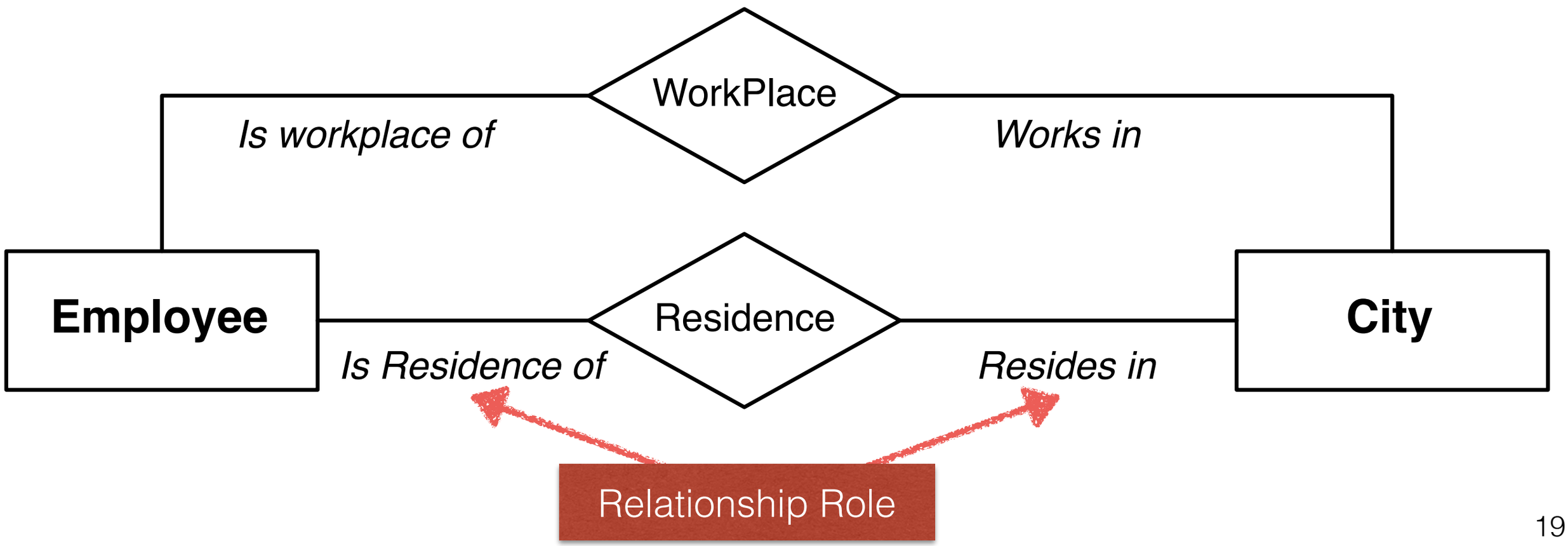
- **Logical links** between two or more entity types
  - Defines a set of associations among occurrences from these entity types
  - An entity type is said to *participate* in a relationship



- Examples:
  - **`RESIDENCE`** is an example of a relationship that can exist between the entity types **`CITY`** and **`EMPLOYEE`**
  - **`EXAM`** is an example of a relationship that can exist between the entity types **`STUDENT`** and **`COURSE`**

Examples of relationship types in the E-R model





About Relationship Types

- An occurrence of a relationship type is an *n-tuple* made up of **occurrences of entity types**, one for each of the entity types involved

- **Degree** of a relationship type
  - Number of participating entity types (*Binary*, T*ernary, Recursive*)

- No identical occurrences!

Structural Constraints of Relationship Types

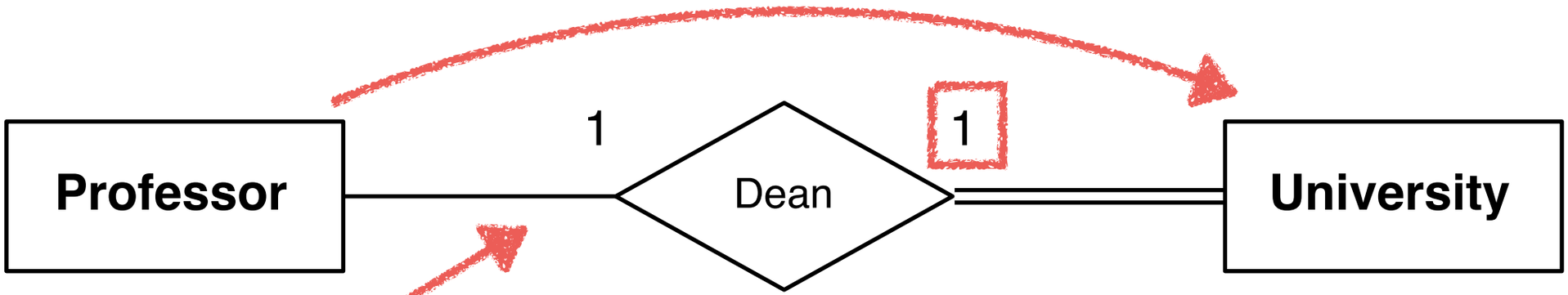
- **Cardinality**: describes the **maximum** and **minimum** number of relationship *occurrences* in which an entity occurrence can participate
  - Specified for each entity participating in a relationship

- `maximum` (**cardinality ratio**) can be
  - `1`: each occurrence of the entity is associated **at most** with a single occurrence of the relationship
  - `N`: each occurrence of the entity is associated with **an arbitrary number** of occurrences of the relationship

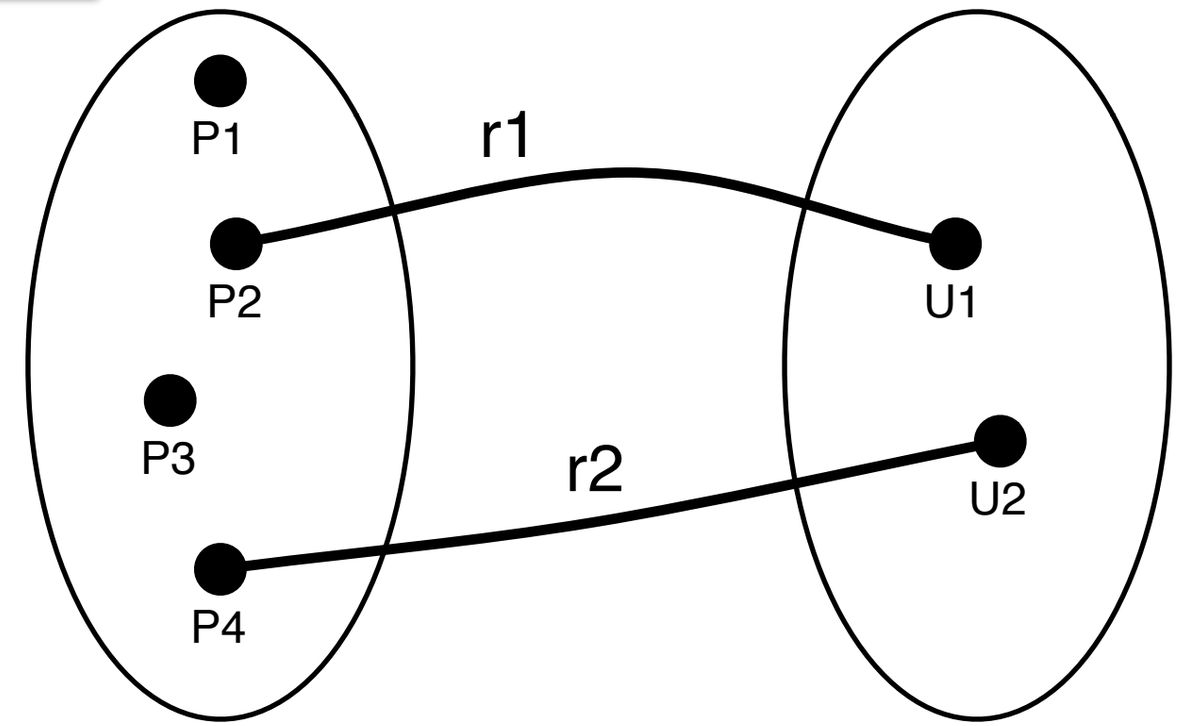
- `minimum` (**participation constraint**) can be
  - `0`: the participation in the relationship is *optional* or **partial**
  - `1`: the participation is *mandatory* or **total** (**existence dependency**)

Example of 1 : 1 relationship type / I

A professor might be the dean of a single university



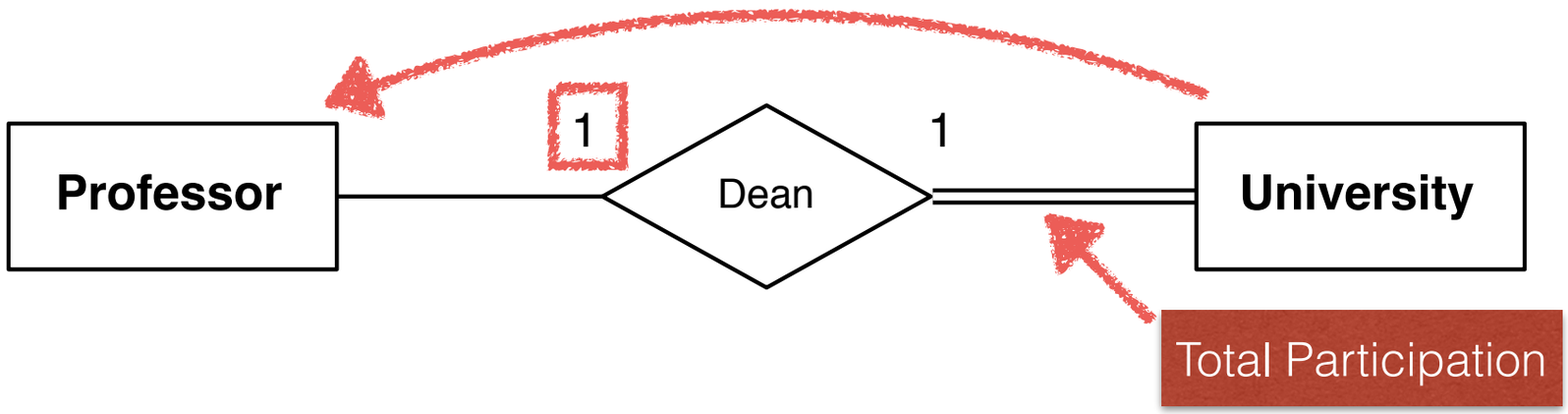
Partial Participation

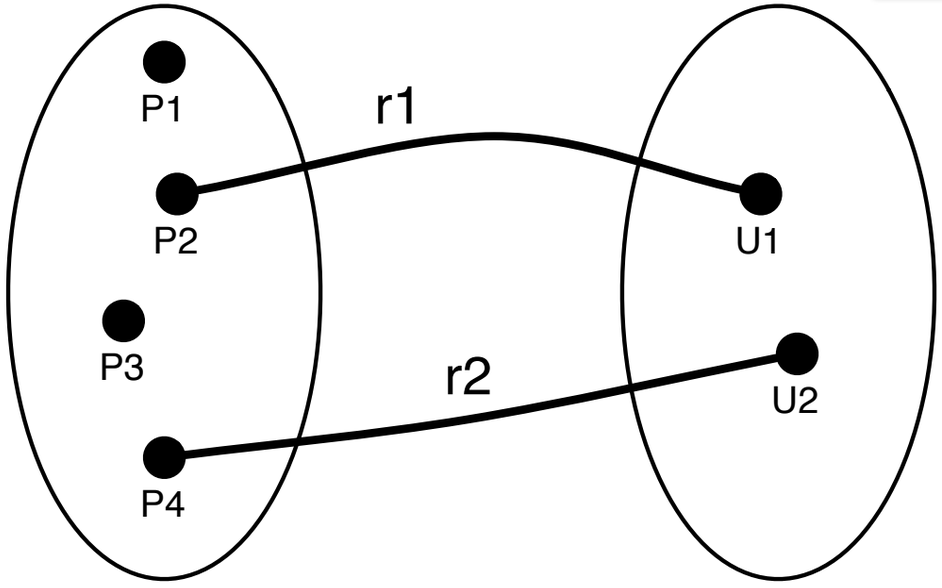


Professor **University**

Example of 1 : 1 relationship type /2

A university must have one professor acting as a dean

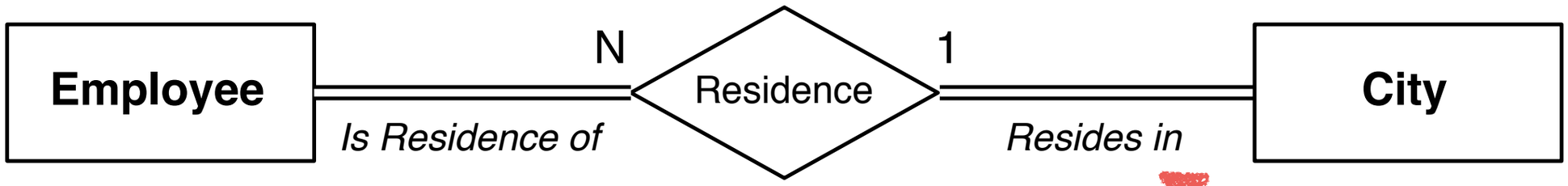




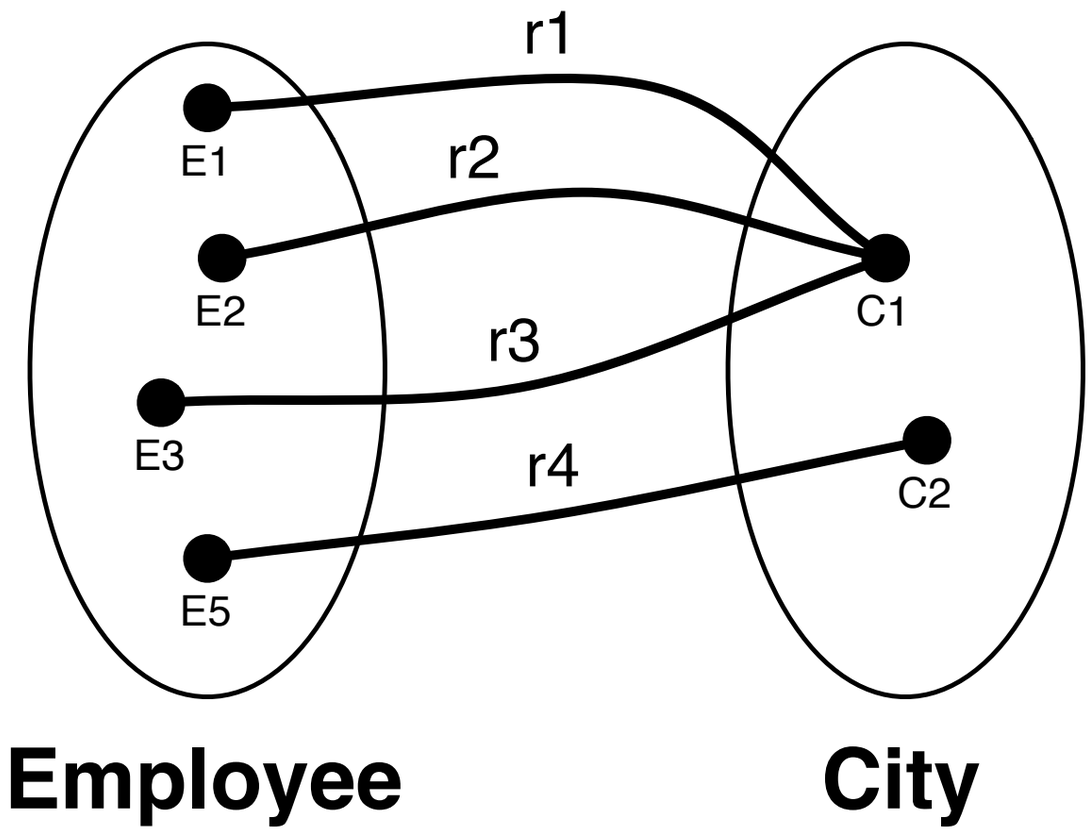
- Minimum cardinalities are rarely 1 for **all** the entities participating in a relationship

Example of 1 : N relationship type

An employ must reside in a city. A city must be the residence of at least one employee

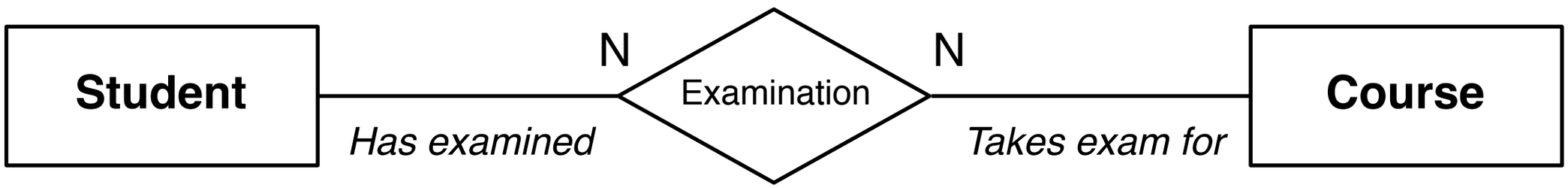


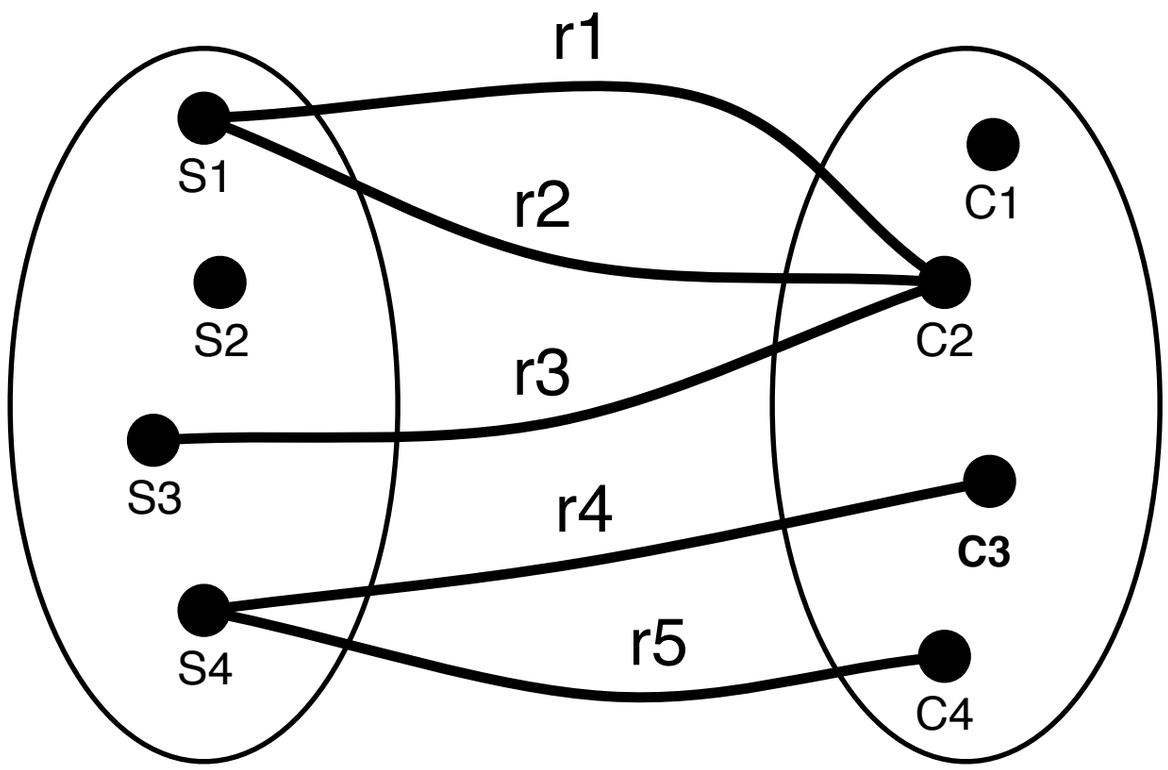
Relationship Role



Example of N : N relationship type

A student can take exams for many courses. A course might examine several students.





Student **Course**

ASQ Exercise (30 sec)

For the binary relationships below, suggest cardinality ratios based on common-sense meaning of the entity types

Student	Teacher

1. one-to-one (1:1)
2. one-to-many (1:N)
3. many-to-one (N:1)
4. many-to-many (N:N)

ASQ Exercise (30 sec)

For the binary relationships below, suggest cardinality ratios based on common-sense meaning of the entity types

ClassRoom	Wall

1. one-to-one (1:1)
2. one-to-many (1:N)
3. many-to-one (N:1)
4. many-to-many (N:N)

ASQ Exercise (30 sec)

For the binary relationships below, suggest cardinality ratios based on common-sense meaning of the entity types

Country	CurrentPresident

1. one-to-one (1:1)
2. one-to-many (1:N)
3. many-to-one (N:1)
4. many-to-many (N:N)

ASQ Exercise (30 sec)

For the binary relationships below, suggest cardinality ratios based on common-sense meaning of the entity types

Class	Instructor

1. one-to-one (1:1)
2. one-to-many (1:N)
3. many-to-one (N:1)
4. many-to-many (N:N)

ASQ Exercise (30 sec)

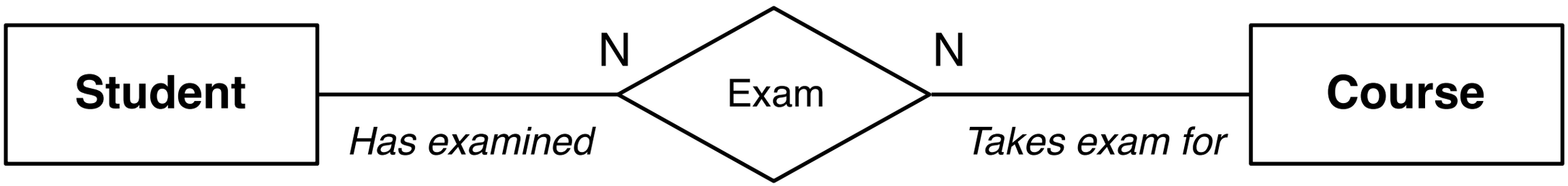
For the binary relationships below, suggest cardinality ratios based on common-sense meaning of the entity types

E-bay Auction item	E-bay bid

1. one-to-one (1:1)
2. one-to-many (1:N)
3. many-to-one (N:1)
4. many-to-many (N:N)

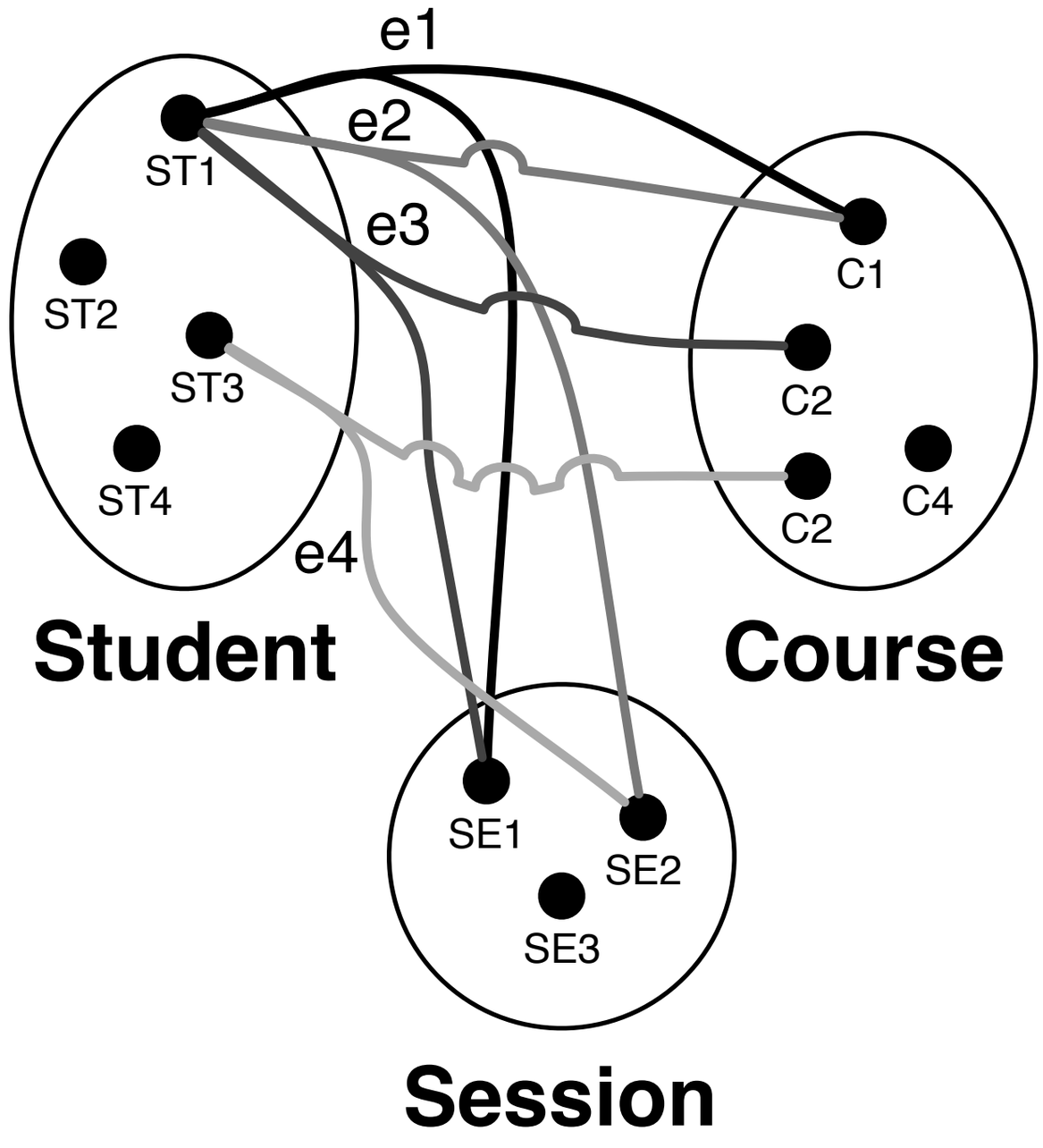
Limits of binary relationships

- Can we represent a student taking the same exam multiple times/sessions?

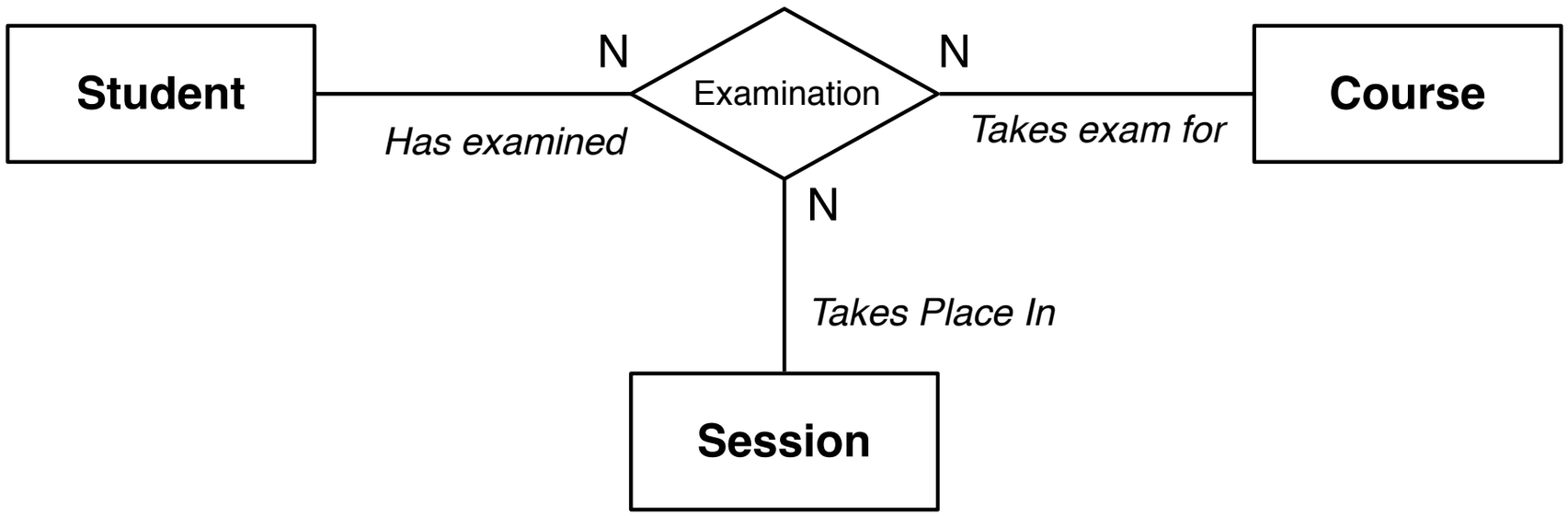


Example of ternary relationship

- A student can repeat the same exam in multiple sessions
- Example:
  - `ST1 C1 SE1`
  - `ST1 C1 SE2`
  - `ST1 C2 SE1`



Example of ternary relationship type



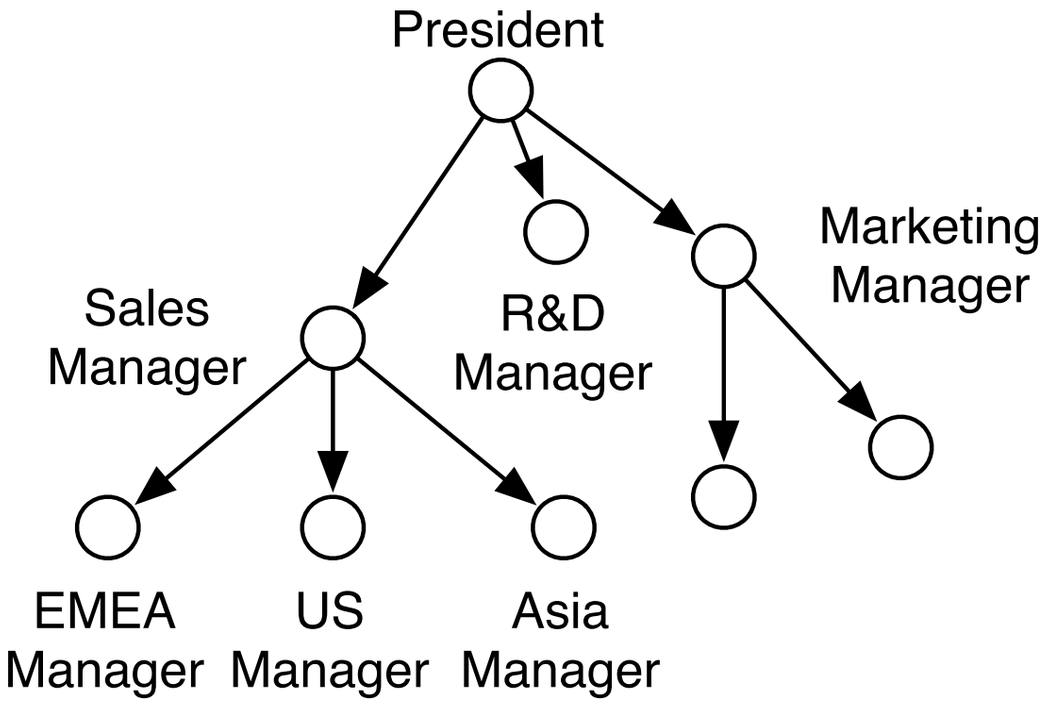
- Maximum cardinalities in an n-ary relationships are almost always `N`

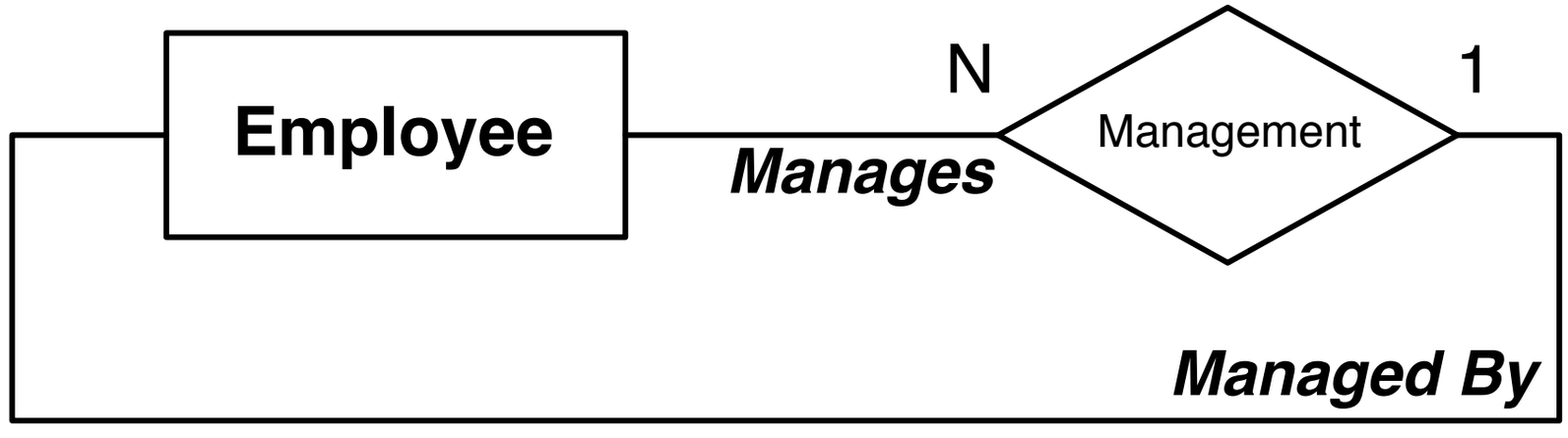
- If an entity `E` participates with maximum cardinality `1`, it is possible to remove the n-ary relationship and relate `E` with one of the other entities with a binary relationship

More in Section 7.9 of the Database Book

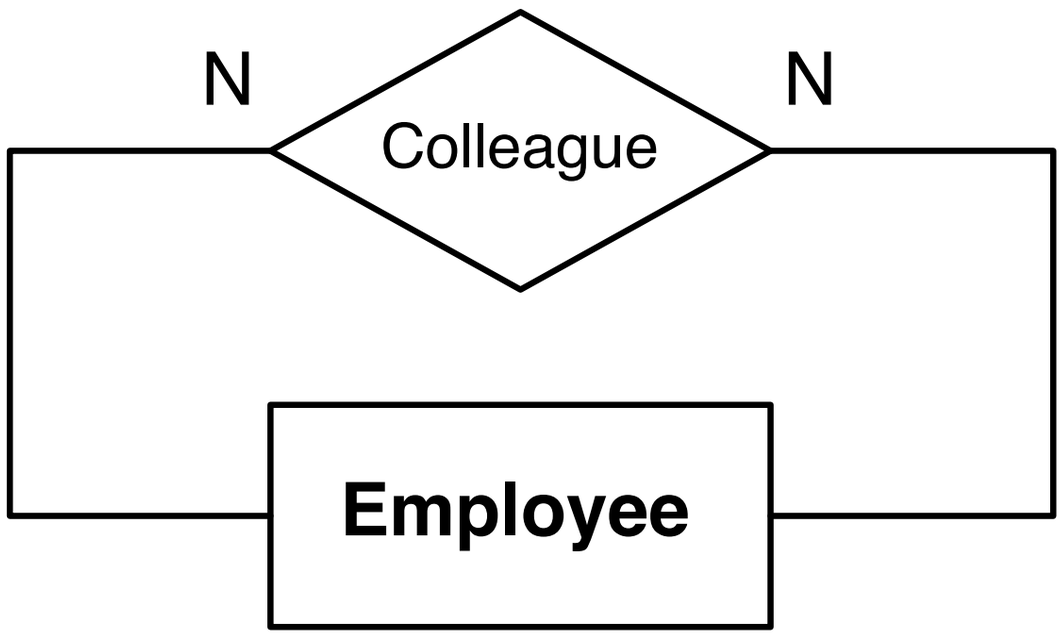
Recursive Relationship Types

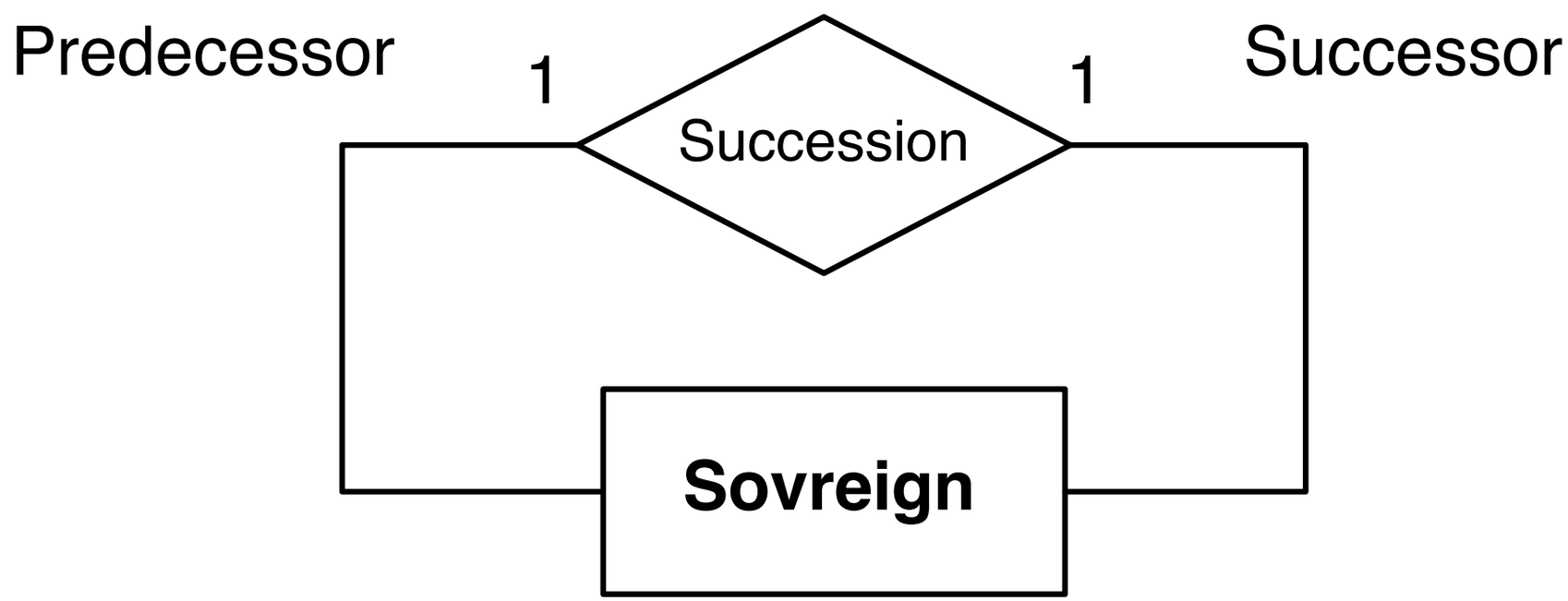
- Relationship where the same entity participates with different roles





Examples of Recursive Relationships Types

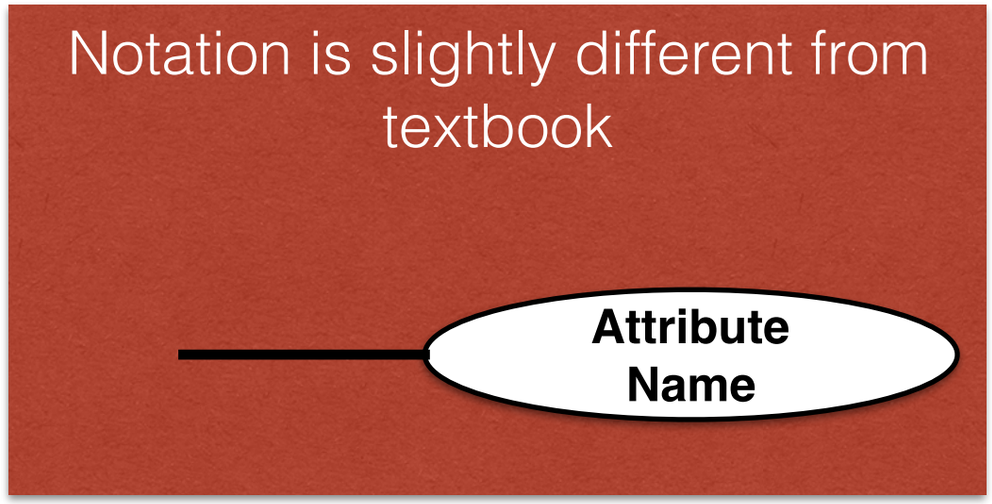




Attributes

Simple Attributes

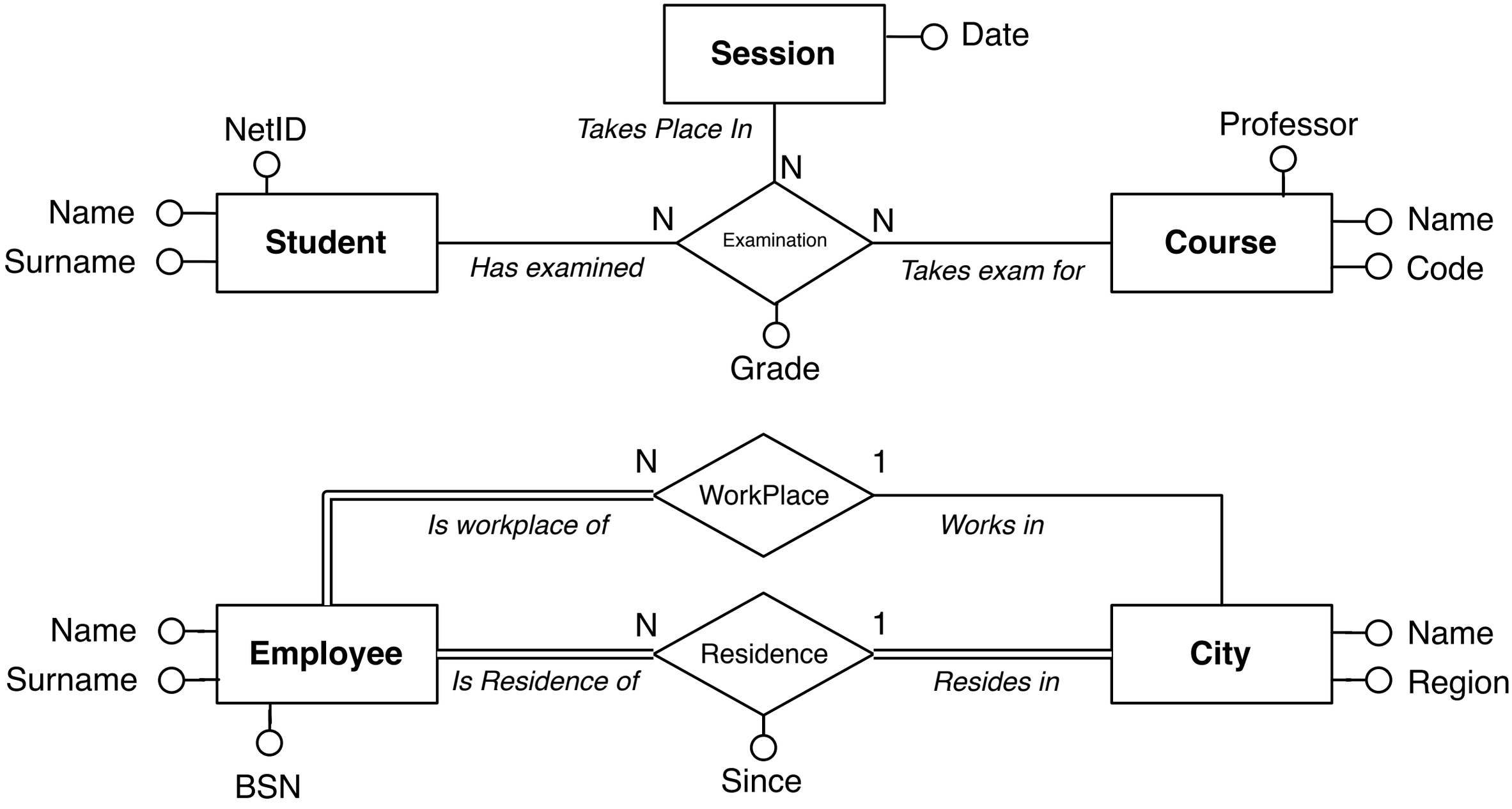
Attribute Name



- Describe the *elementary properties* of **entities** or **relationships**
  - *Surname*, *Salary* and *Age* are possible attributes of the **`EMPLOYEE`** entity
  - *Score* is possible attributes for the relationship **`EXAMINATION`** between **`STUDENT`** and **`COURSE`**

- Each attribute is characterised by a **domain** (not visually represented)

Examples With Attributes



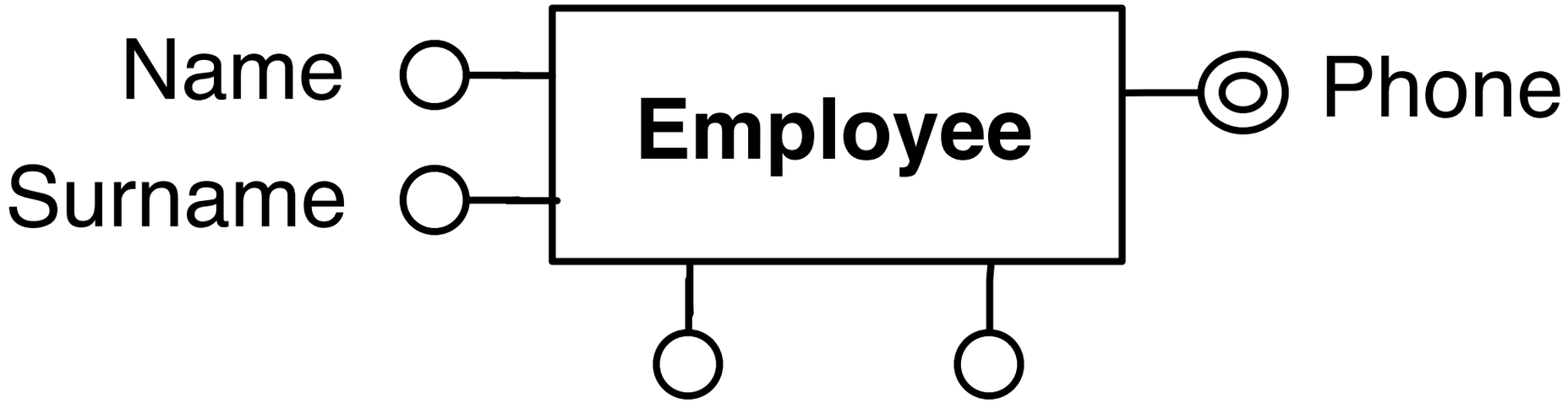
Attribute Cardinality

- Describes the **minimum** and **maximum** number of values of the attribute associated with each occurrence of an entity or a relationship
  - Typically, cardinality is equal to `(1,1)` and is omitted

- Minimum cardinality `0` means that the value can be `NULL`

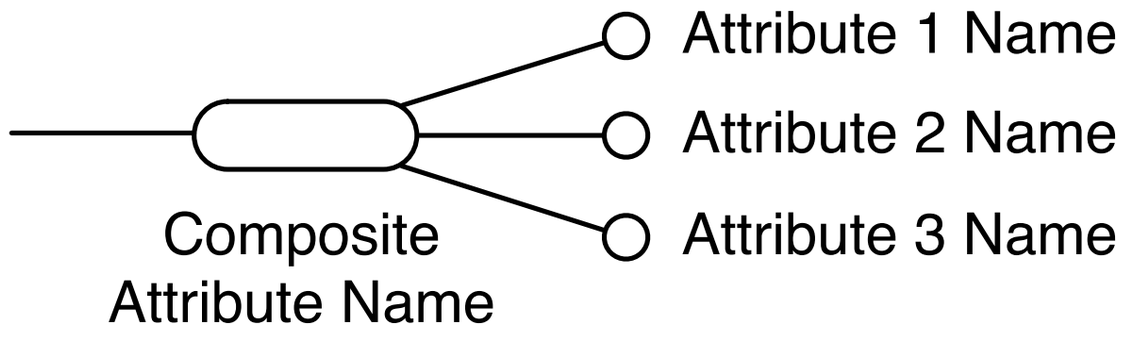
- Maximum cardinality `N` means that the attribute may assume more than one value in the same instance
  - In this case we talk about **multi-valued** attributes and they can be represented as follows

Example

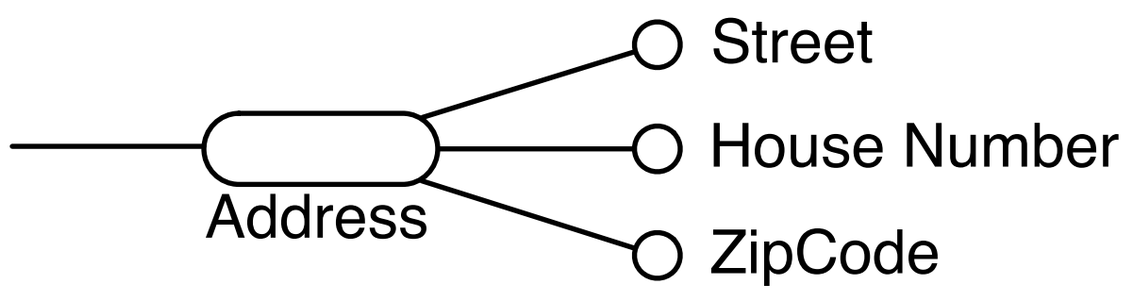


- Beware of multivalued attributes!
  - They might represent situations that could be modelled with additional entity and relationship types
  - E.g. Job as multivalued attribute vs. entity

Composite Attributes



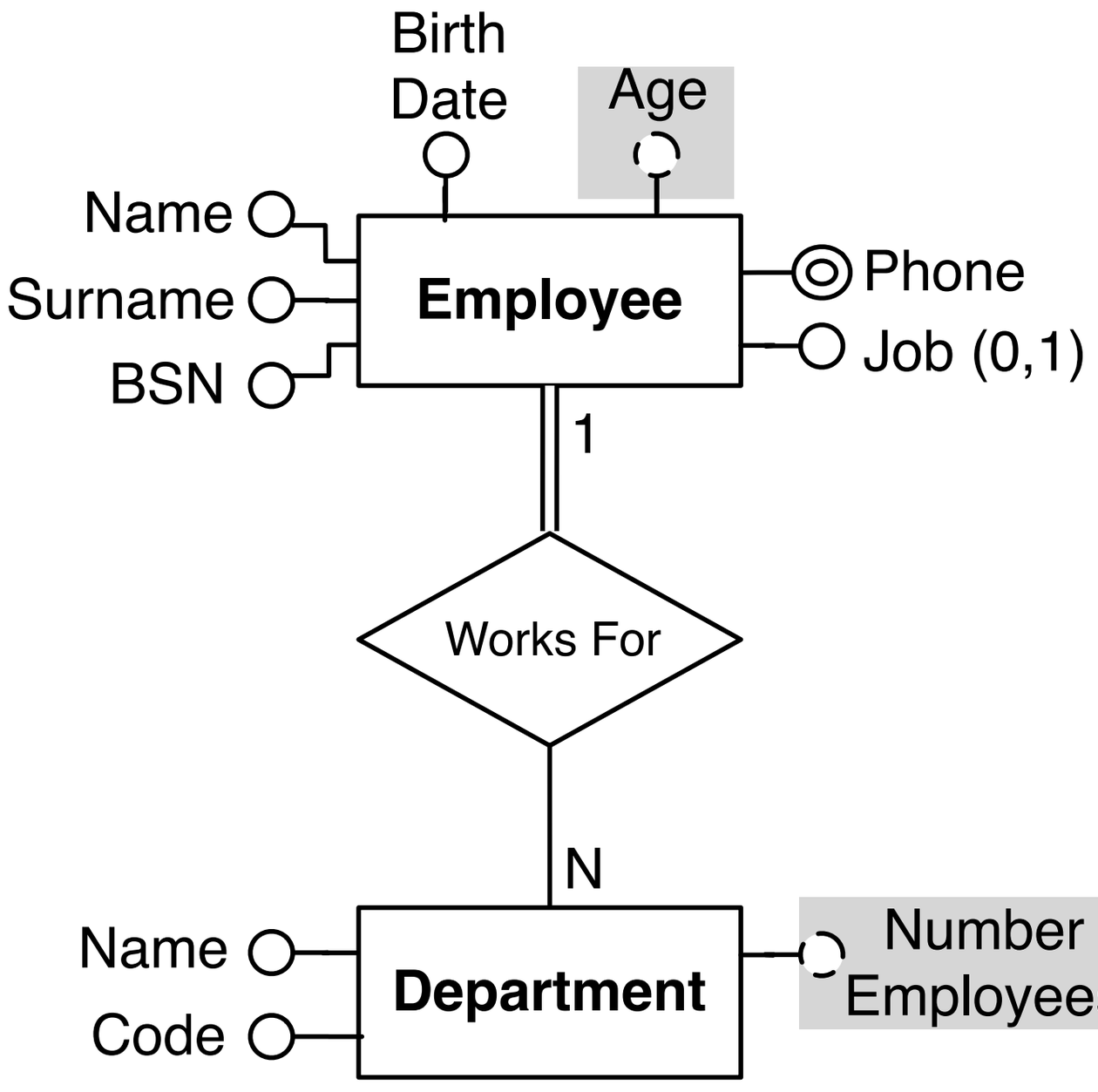
- Achieved by concatenating simpler attribute types
- Pictured by trees of atomic attributes



- Composite attributes can form hierarchies

Derived Attributes

- Attributes having values are generated from other attributes
  - calculations, algorithms or procedures
- Examples
  - **`Age`** from **`BirthDate`**
  - **`Number of Employee`** in a department



Attributes of Relationship Types

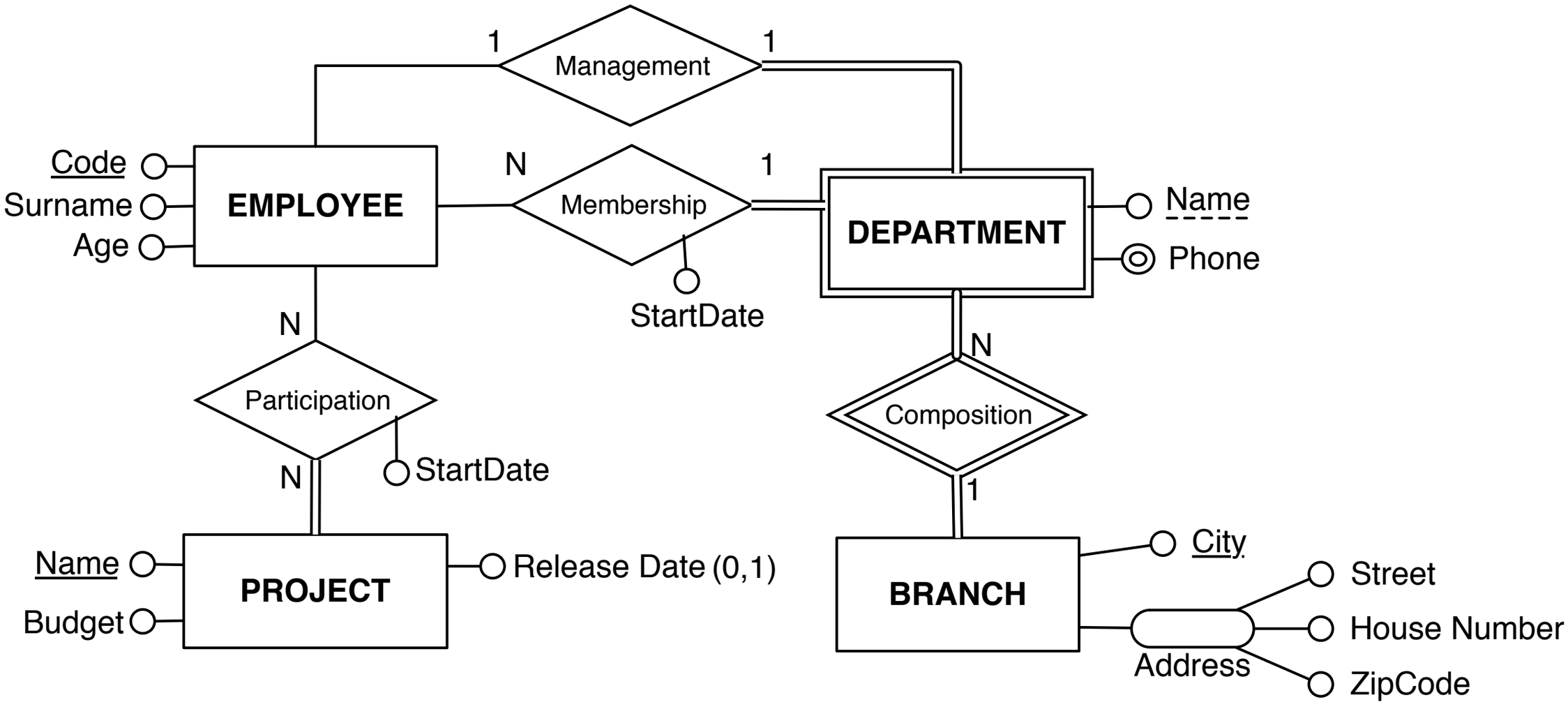
- Attributes of `1:1` relationship types can be migrated to one entity type

- Attributes of `1:N` relationship type can be migrated only to entity type on `N`-side of relationship

- In **`N:N`** relationship types, some attributes may be determined by combination of participating entities
  - Must be specified as relationship attributes
  - E.g. **`Grade`** for **`STUDENT`** and **`EXAMINATION`**

ASQ Exercise (45 sec)

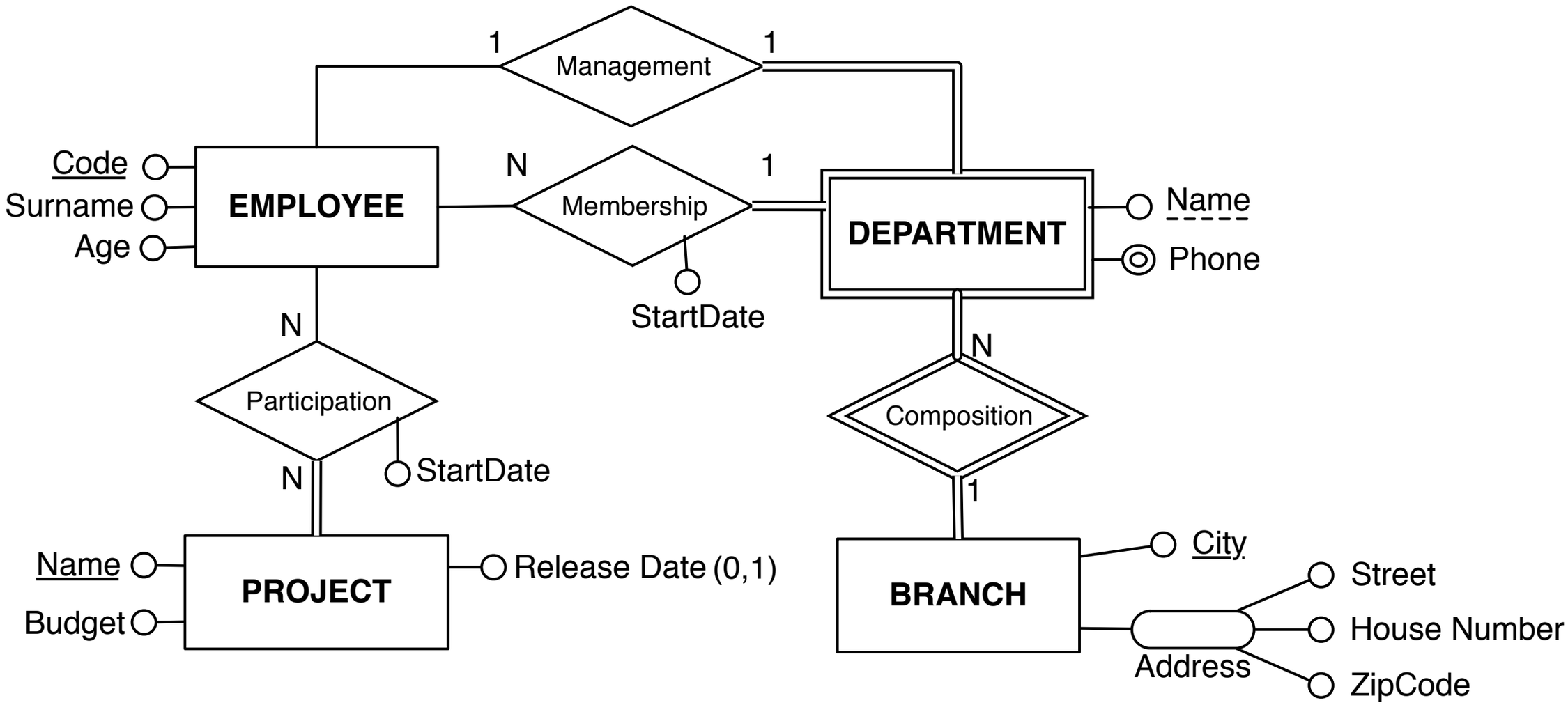
Consider the model below. How many multi-valued attributes are there?



1 1 2 2 3 3 3 5

ASQ Exercise (45 sec)

Consider the model below. To which entity or relationship would you add an attribute “Role in project”



1 Employee

2 Participation

3 Project

ASQ Exercise (45 sec)

Which of the following statements about the attributes cardinality is correct?

1. An attribute cardinality equal to (1,1) can be omitted as the presence of an attribute value is optional

2. Multivalued attributes represent situations that can be often modelled with additional entities.

3. The maximum cardinality of an attribute cannot be an exact number (e.g. 4)

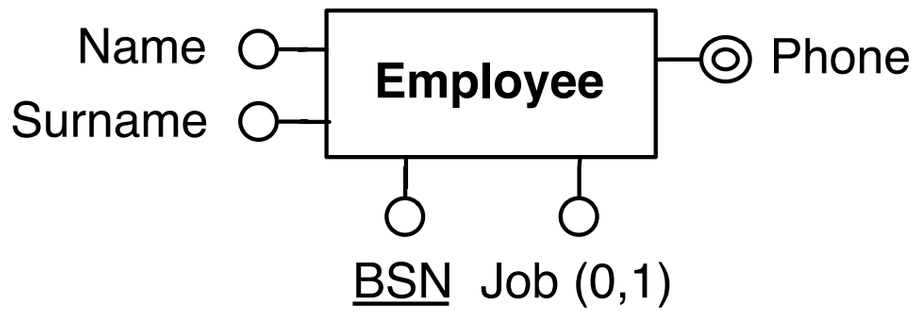
Identifiers

Identifiers

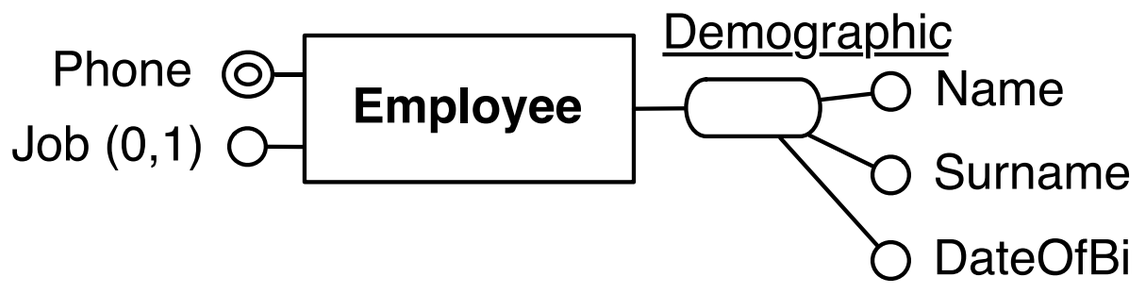
- Describe the concepts of the schema that allow the **unambiguous identification** of the entity instances
  - Attributes **and/or** entities
- They are specified for each entity of a model
- Relationships have no identifier!

Internal Identifiers

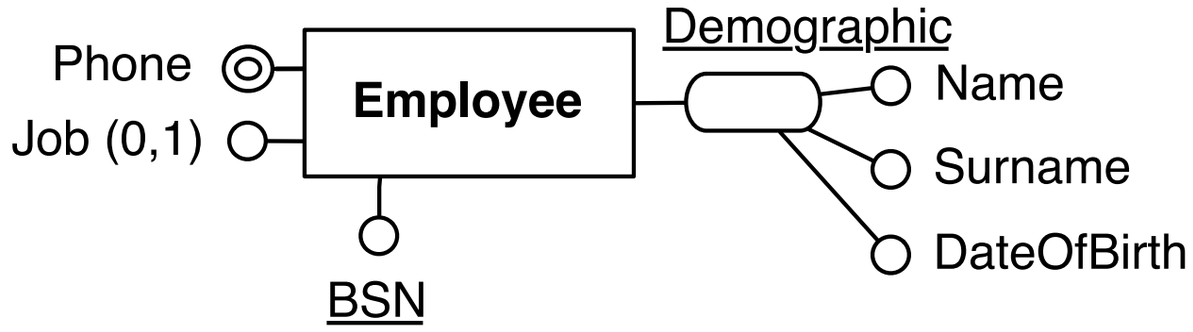
- An identifier is often formed by one or more attributes of the entity itself
- In this case we talk about an **internal identifier**
  - Also known as a **key**



- An identifier can involve one or more attributes
  - Each of them has (1,1) cardinality
  - A new <u>composite attribute</u> becomes key



- Each entity must have one identifier, but can have *more than one*
  - Then *each distinct underlined attribute is a key*
  - There is **no primary key** in ER (only keys)

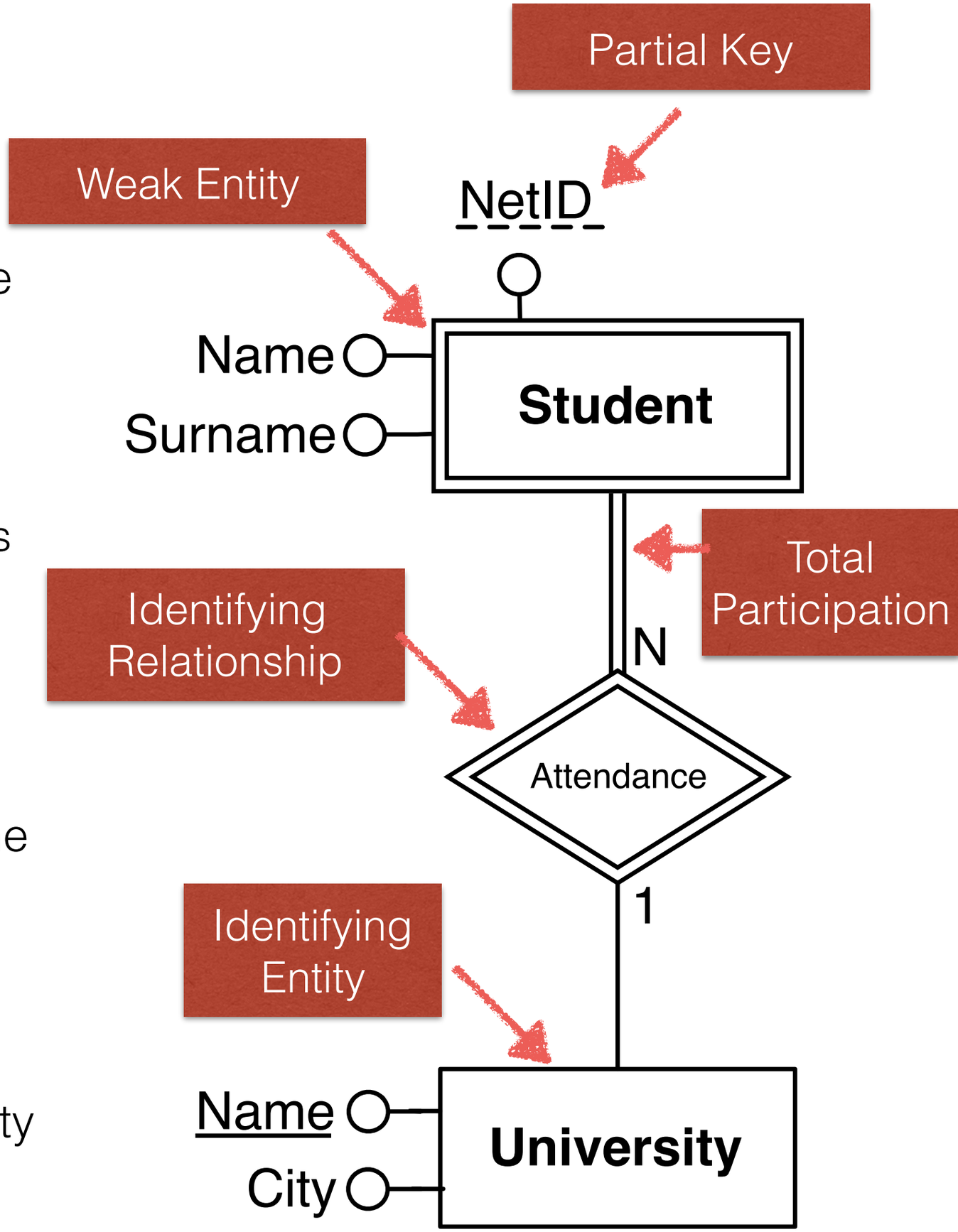


Weak Entities

- Sometimes the attributes of an entity are not sufficient to identify its occurrences unambiguously
  - **Weak Entities**
  - Entities may not have internal identifiers
  - Or partial identifiers **(Partial keys)**

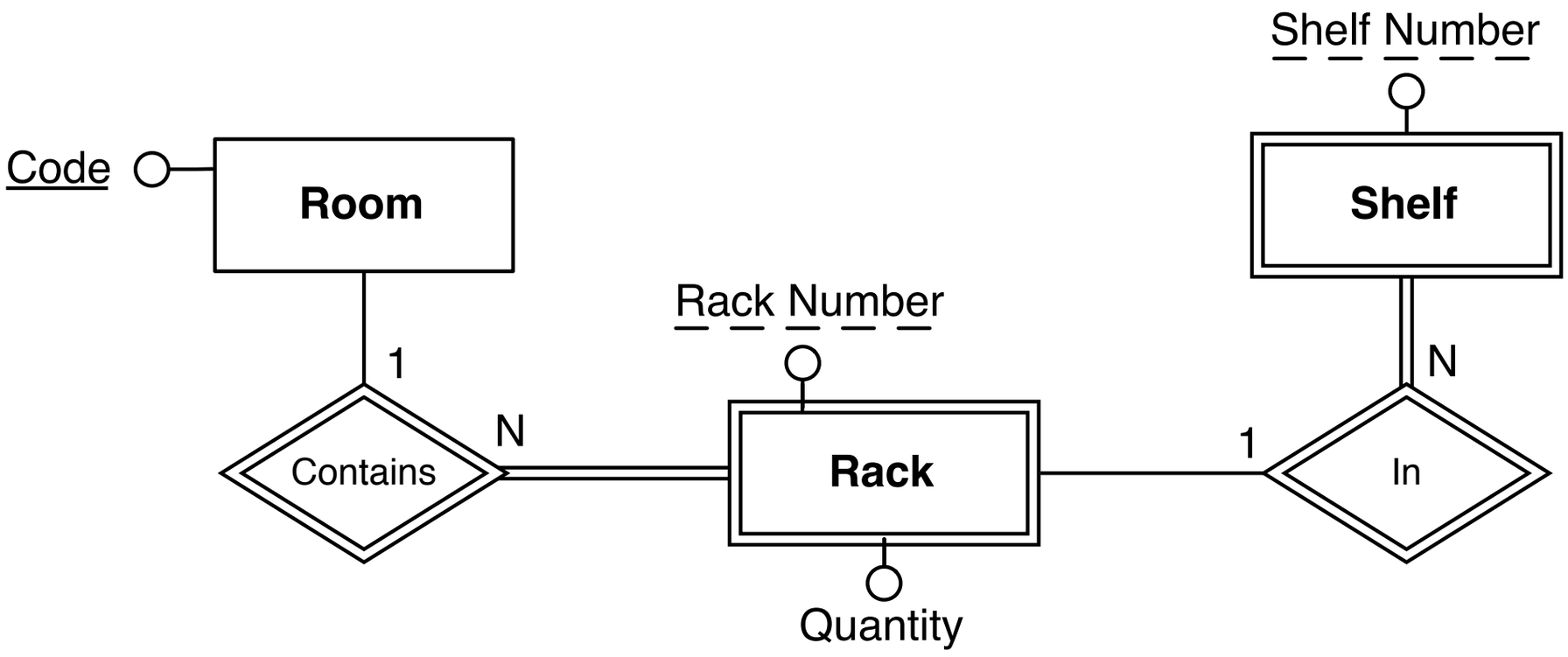
- **Other entities** need to be involved in the identification
  - This is called an **external identifier**

- The relationship that relates a weak entity to its owner is called **identifying relationship**



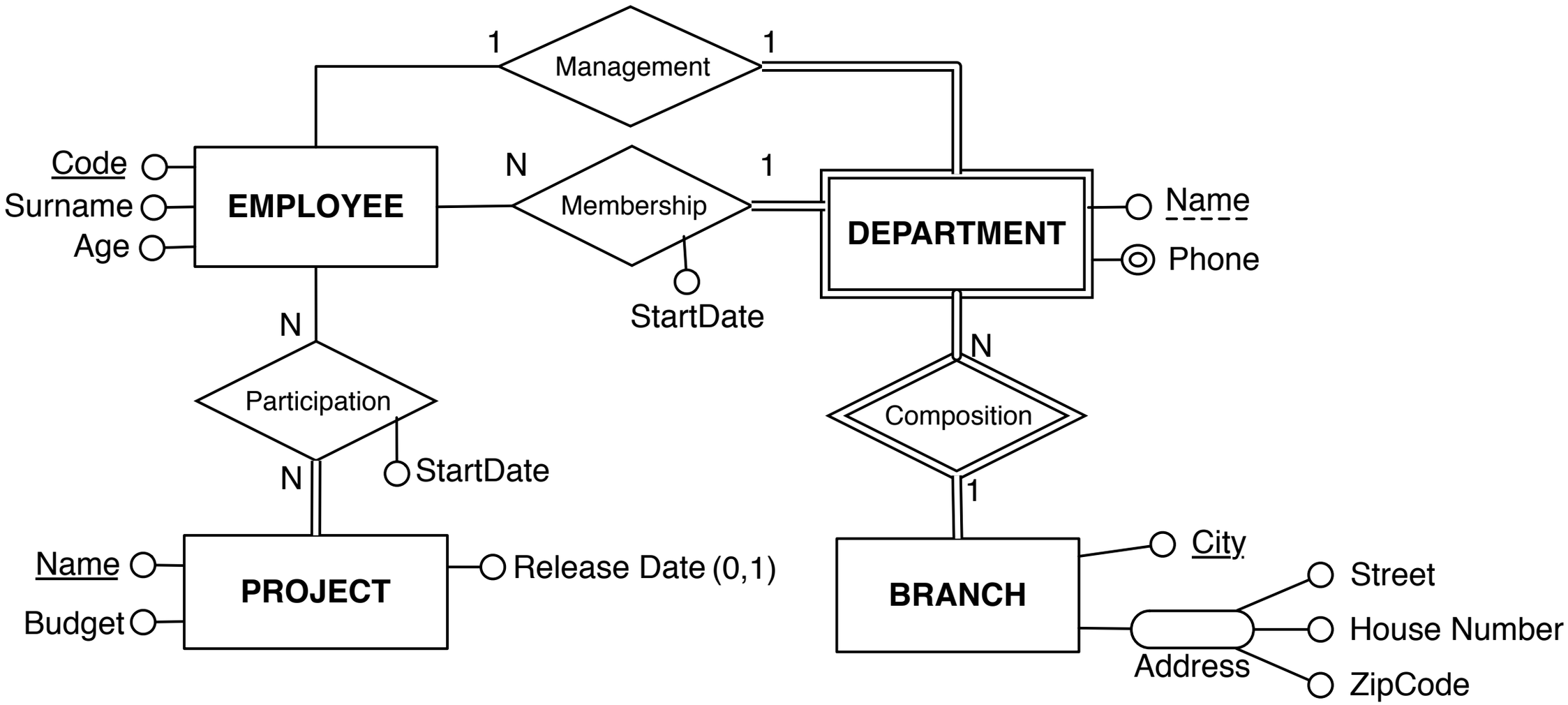
General Observations

- Weak entities can sometimes be represented as complex attributes
- An external identifier can involve one or more entities, provided that each of them is member of a relationship to which the entity to identify participates with cardinality equal to (1,1)
- An external identifier can involve an entity that is in its turn identified externally, as long as cycles are not generated



ASQ Exercise (1 min)

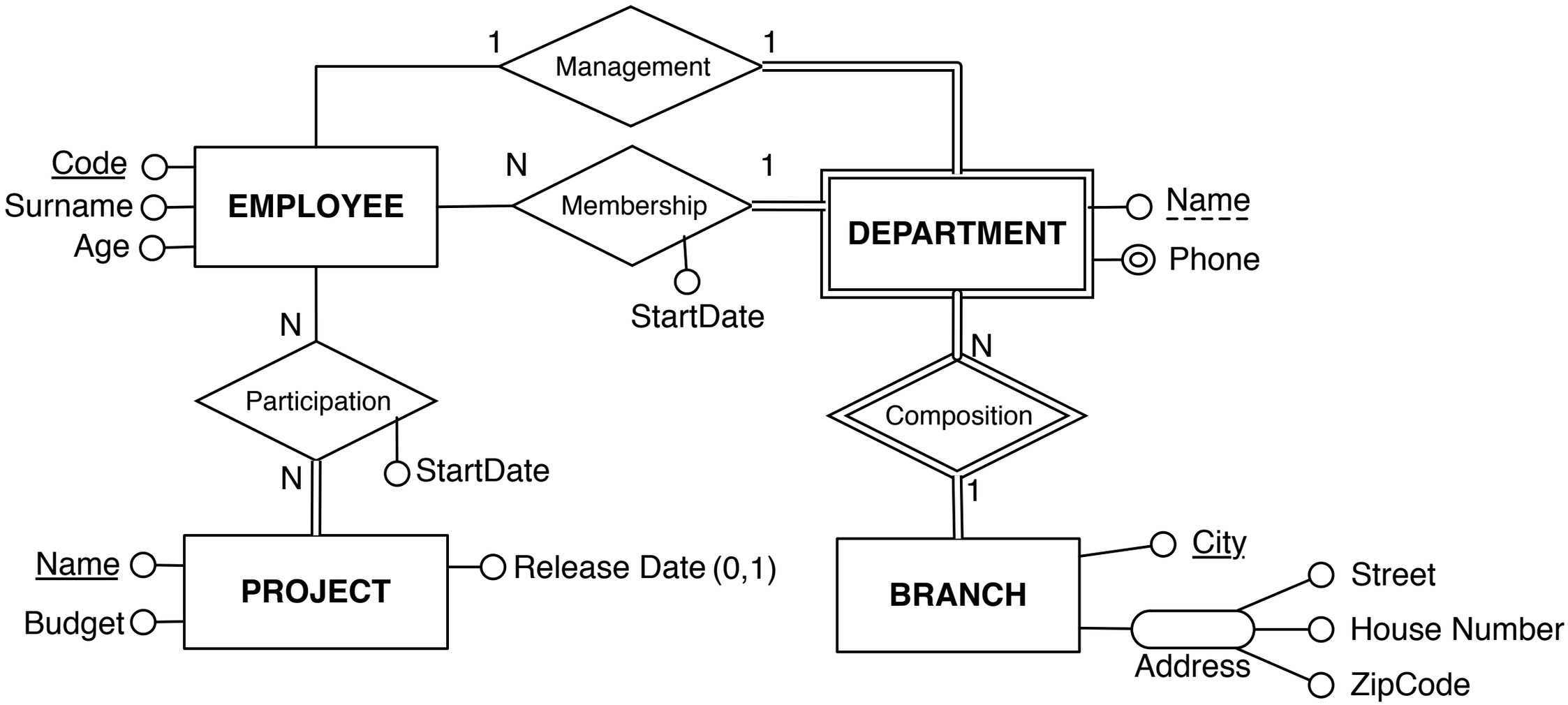
Consider the EER diagram in figure. How many internal identifiers are there?



1 2 2 3 3 4 4 6

ASQ Exercise (1 min)

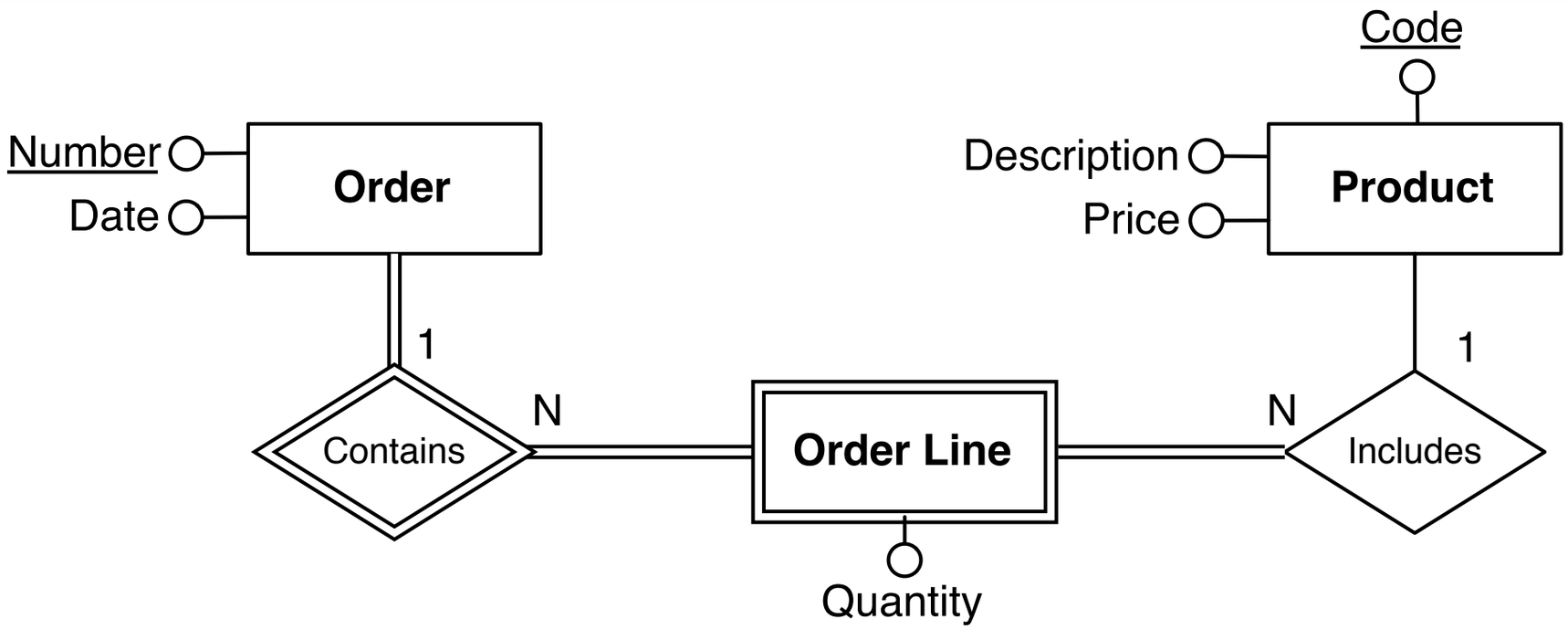
Consider the EER diagram in figure. Which entity type is an **identifying entity type**?



1. Project
2. Employee
3. Department
4. Branch

ASQ Exercise (1 min)

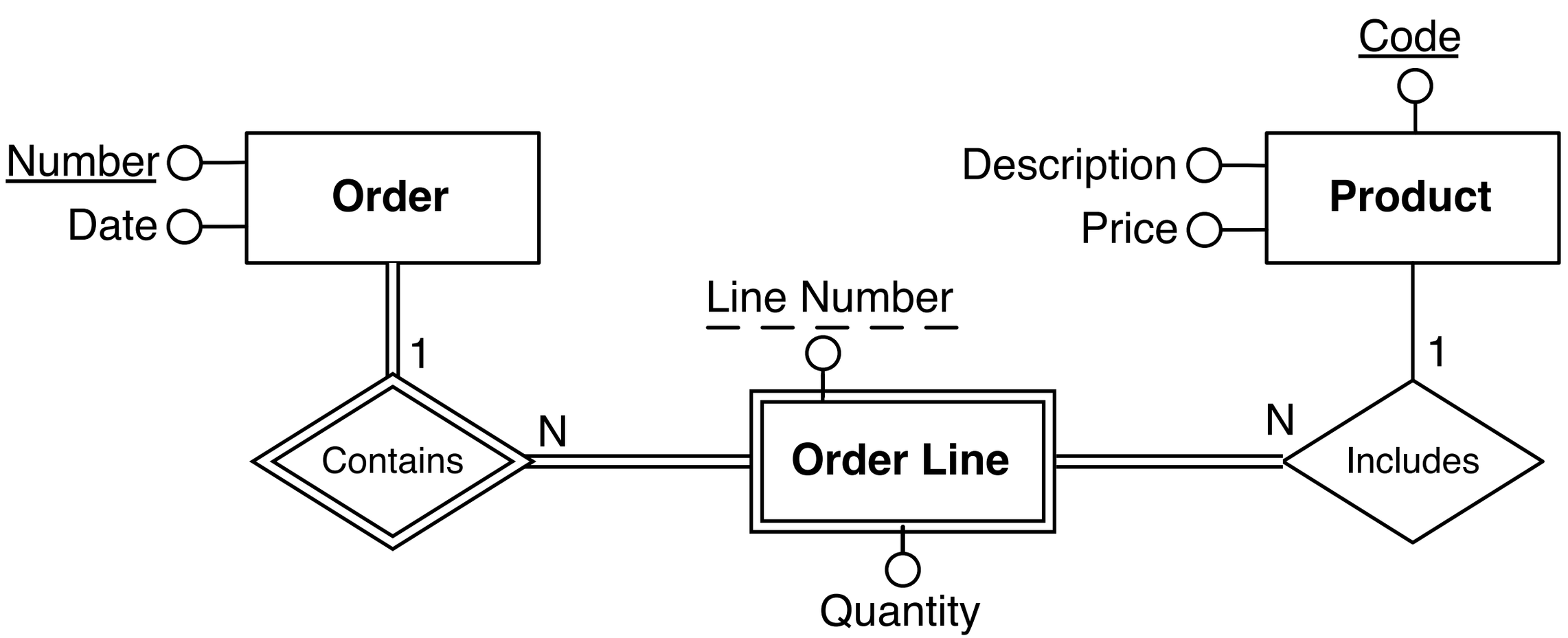
Consider the EER diagram in figure. How can it be extended to allow in the same order, multiple order lines for the same product?



1. Add "LINE NUMBER" attribute to "Order Line"
2. Add "LINE NUMBER" attribute to "Contains"
3. Add "LINE NUMBER" attribute to "Includes"
4. It is Impossible

SOLUTION

Line Number is a partial key



Readings

- **Database Book**
  - *Chapter **7** of the Database book*

- **Suggested readings on Blackboard**

End of Lecture