

TRANSFERABILITY

P R E P A R E D B Y : L U I S M E I N G

OBJECTIVES:

01

Relationship Transferability

02

Relationship Types

03

Redundant Relationships

04

**Resolving Many-to-Many
Relationships**

05

CRUD Analysis



Relationship Transferability

- this is where an entity's relationship can be transferred to another instance of the other entity
- where relationships between instances can be switched to another

Relationship Transferability

- is based on the optionality and cardinality of the relationship
 - is also based on real-life business logic
 - is implied by both structural and procedural rules

1.1 Relationship Transferability



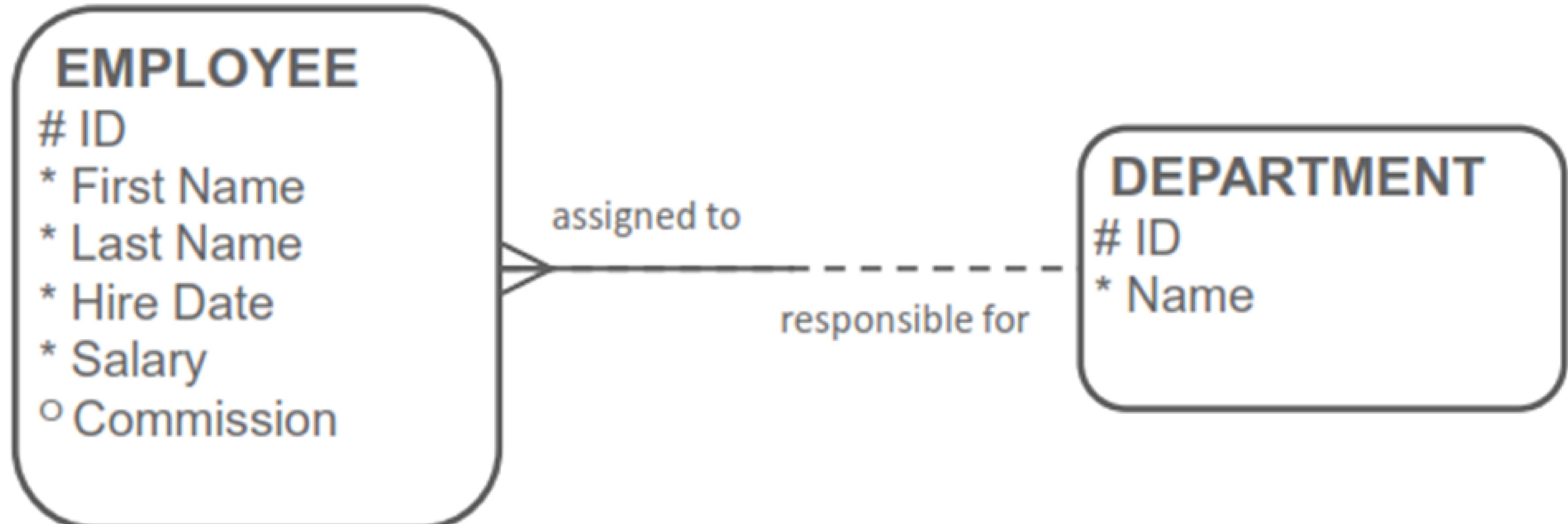
example 1

- In a company database, every employee should always be in a department. A department can be responsible for a lot of employees.

1.1 Relationship Transferability



example 1



1.1 Relationship Transferability

example 1: Optionality

- Must each EMPLOYEE be assigned to a DEPARTMENT?
- Must each DEPARTMENT be responsible for an EMPLOYEE?



example 1: Cardinality

- How many EMPLOYEES can a DEPARTMENT be responsible for?
- How many DEPARTMENTS can an EMPLOYEE be assigned to?

1.1 Relationship Transferability

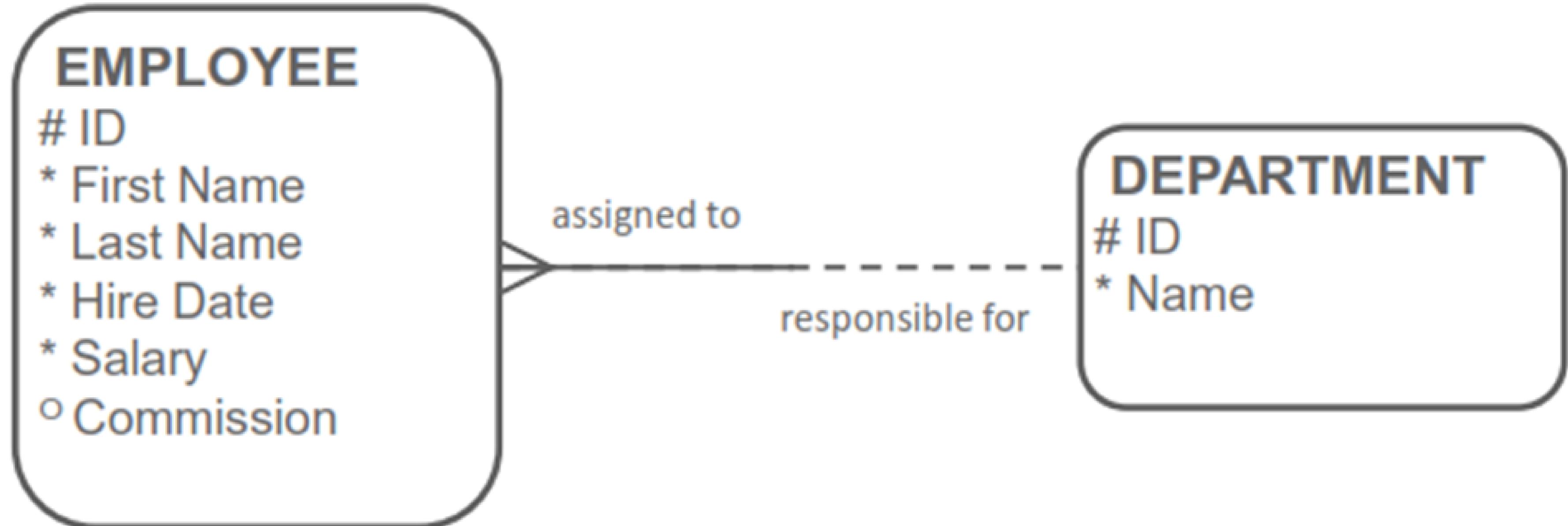
example 1: Transferability

- Can each EMPLOYEE's relationship be transferred to another DEPARTMENT?
- Can each DEPARTMENT's relationship be transferred to another EMPLOYEE?

1.1 Relationship Transferability



example 1





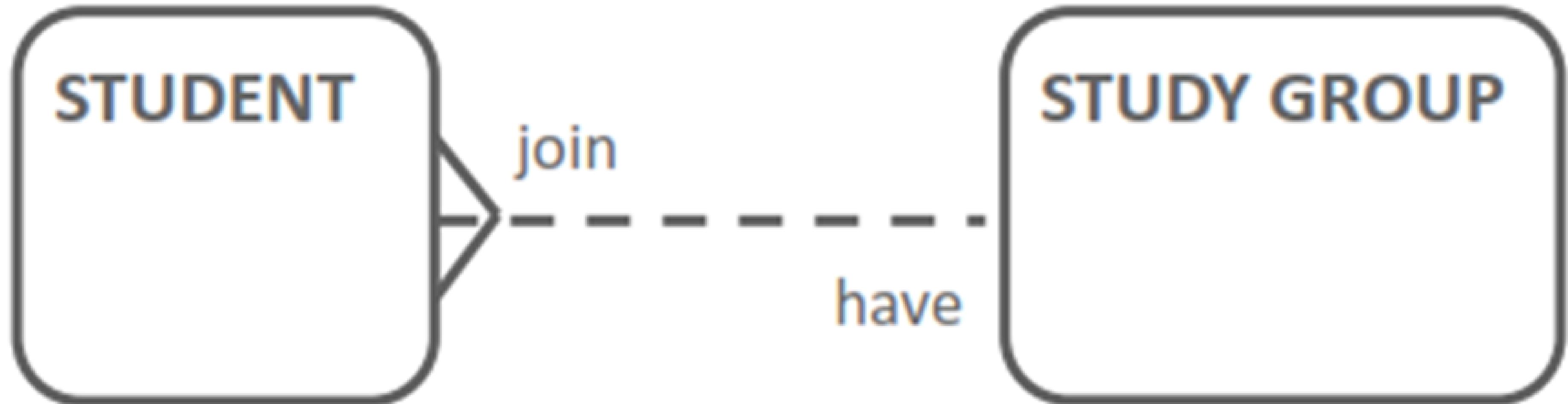
example 2

- A teacher assigned students to different study groups. They created a database to keep track of each student. A student has the option to not join one and the study groups may have multiple students.

1.1 Relationship Transferability



example 2

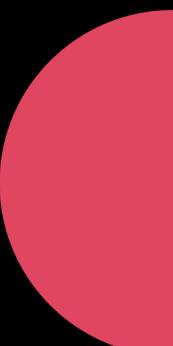


Transferable Relationship



example 2: Optionality

- Must each STUDENT join a STUDY GROUP?
- Must each STUDY GROUP have a STUDENT?





example 2: Cardinality

- How many STUDENTs can a STUDY GROUP have?
- How many STUDY GROUPs can a STUDENT join?

1.1 Relationship Transferability

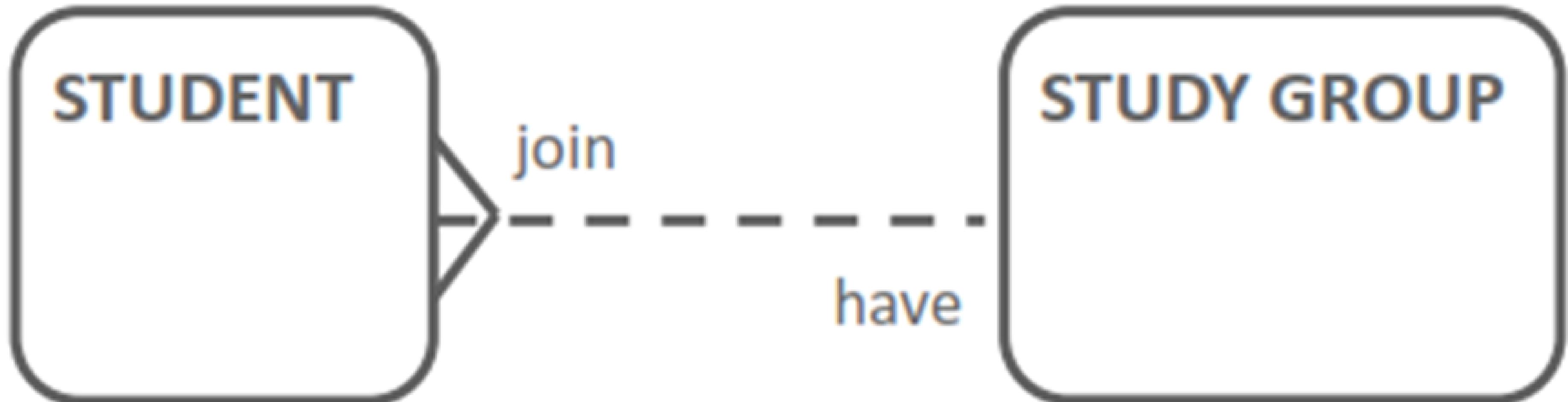
example 2: Transferability

- Can each STUDENT's relationship be transferred to another STUDY GROUP?
- Can each STUDY GROUP's relationship be transferred to another STUDENT?

1.1 Relationship Transferability



example 2



Transferable Relationship

1.1 Relationship Transferability

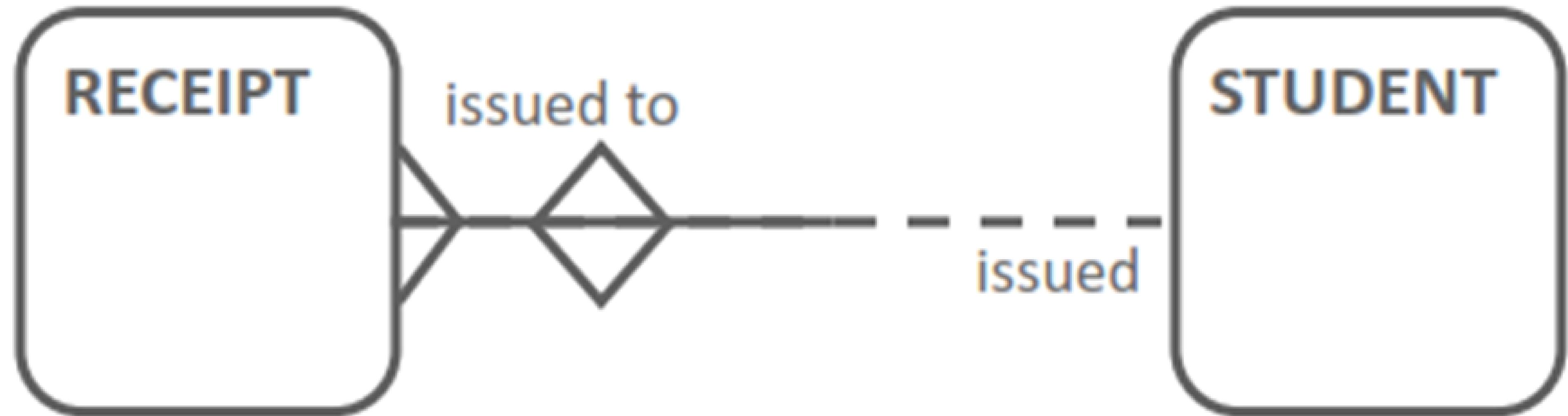


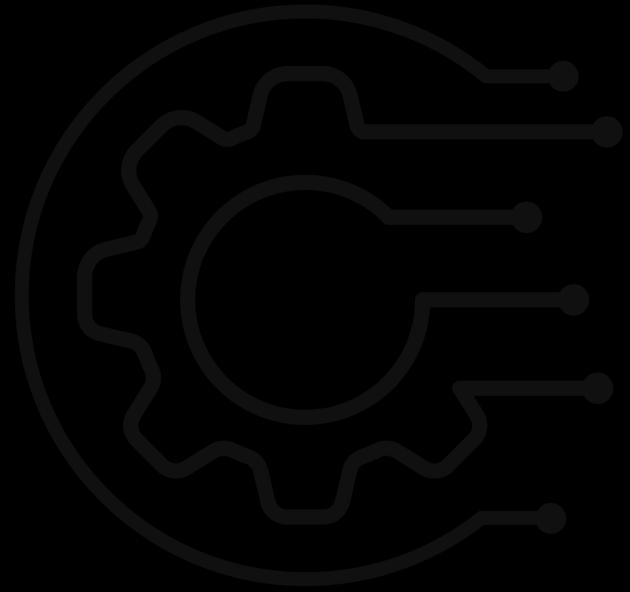
example 3

- A student can be issued a receipt for paying tuition fees, taking a certification exam, or purchasing items at the bookstore.

1.1 Relationship Transferability

example 3





example 3: Optionality

- Must each RECEIPT be issued to a STUDENT?
- Must each STUDENT be issued a RECEIPT?



example 3: Cardinality

- How many RECEIPTS can a STUDENT be issued?
- How many STUDENTS can a RECEIPT be issued to?

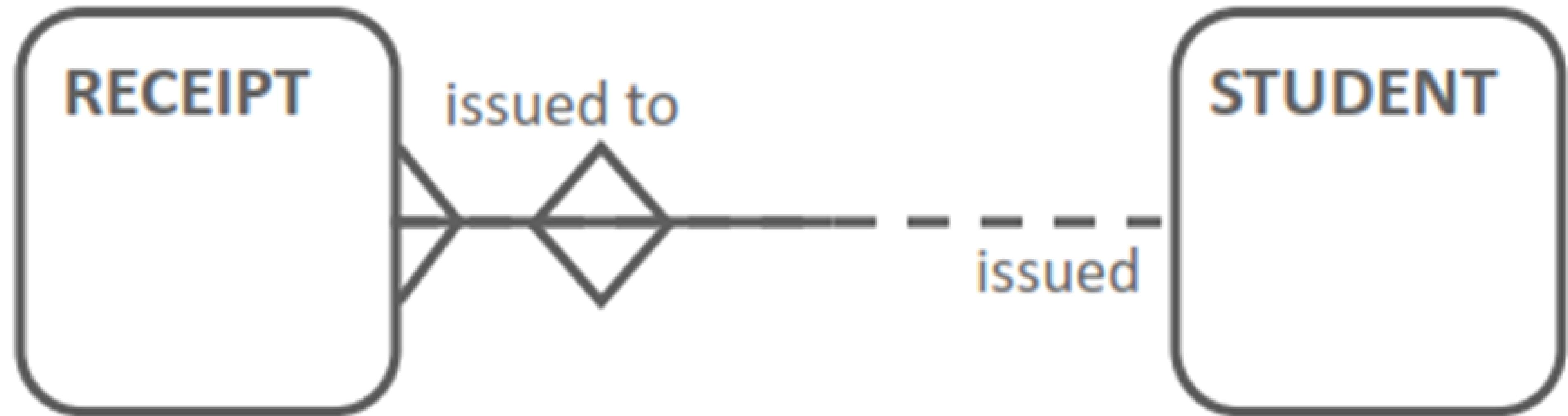
1.1 Relationship Transferability

example 3: Transferability

- Can each RECEIPT's relationship be transferred to another STUDENT?
- Can each STUDENT's relationship be transferred to another RECEIPT?

1.1 Relationship Transferability

example 3





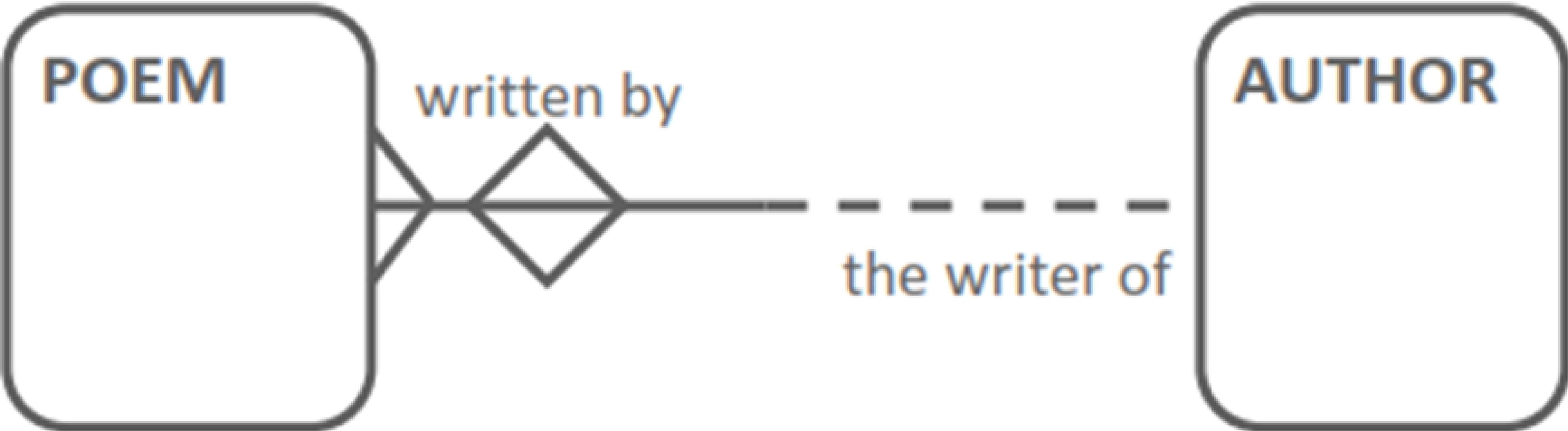
example 4

- The literature copyright committee had created a database to fight plagiarism. Each poem ever publicly written can be traced to an author and each author can write many poems.

1.1 Relationship Transferability



example 4





example 4: Optionality

- Must each POEM be written by an AUTHOR?
- Must each AUTHOR be the writer of a POEM?



example 4: Cardinality

- How many POEMS can an AUTHOR write?
- How many AUTHORS can a POEM have?

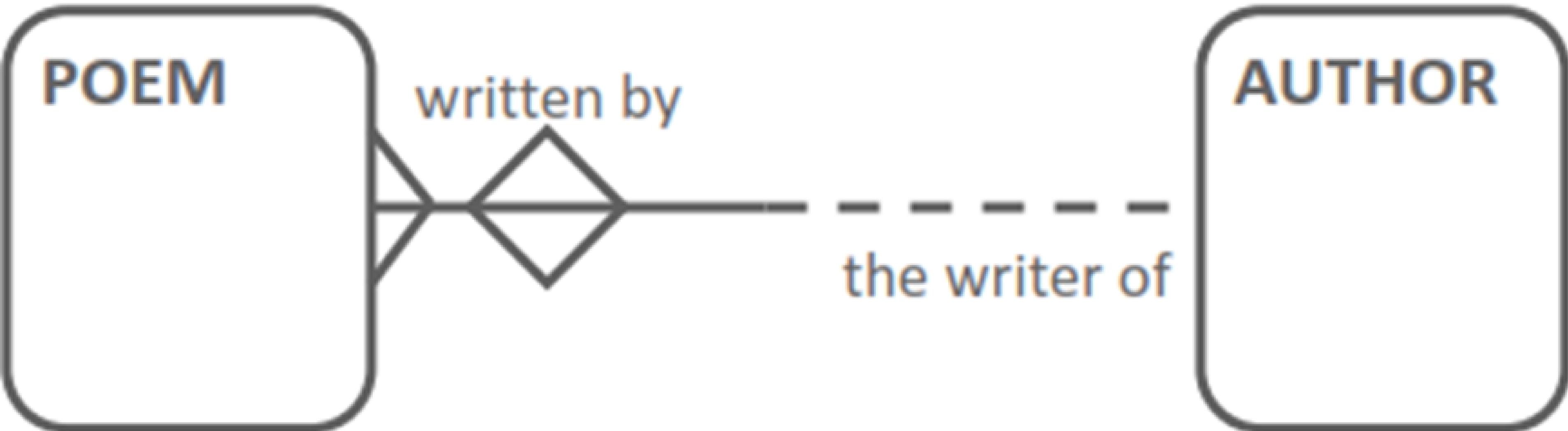
example 4: Transferability

- Can each POEM's relationship be transferred to another AUTHOR?
- Can each AUTHOR's relationship be transferred to another POEM?

1.1 Relationship Transferability



example 4





Conclusion

- Transferability it is where the relationship can or cannot be changed or shifted among entities and their instances



Conclusion

- Take note of the optionality where the non-transferable relationship is placed. It is always at the side where the optionality is mandatory.



Conclusion

- Nontransferable relationships are represented with a diamond on the line



2.1 Relationship Types

Relationship Types

- in terms of cardinality

2.1 Relationship Types

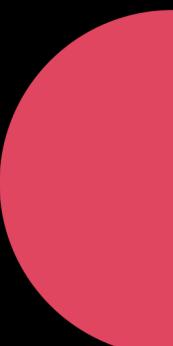
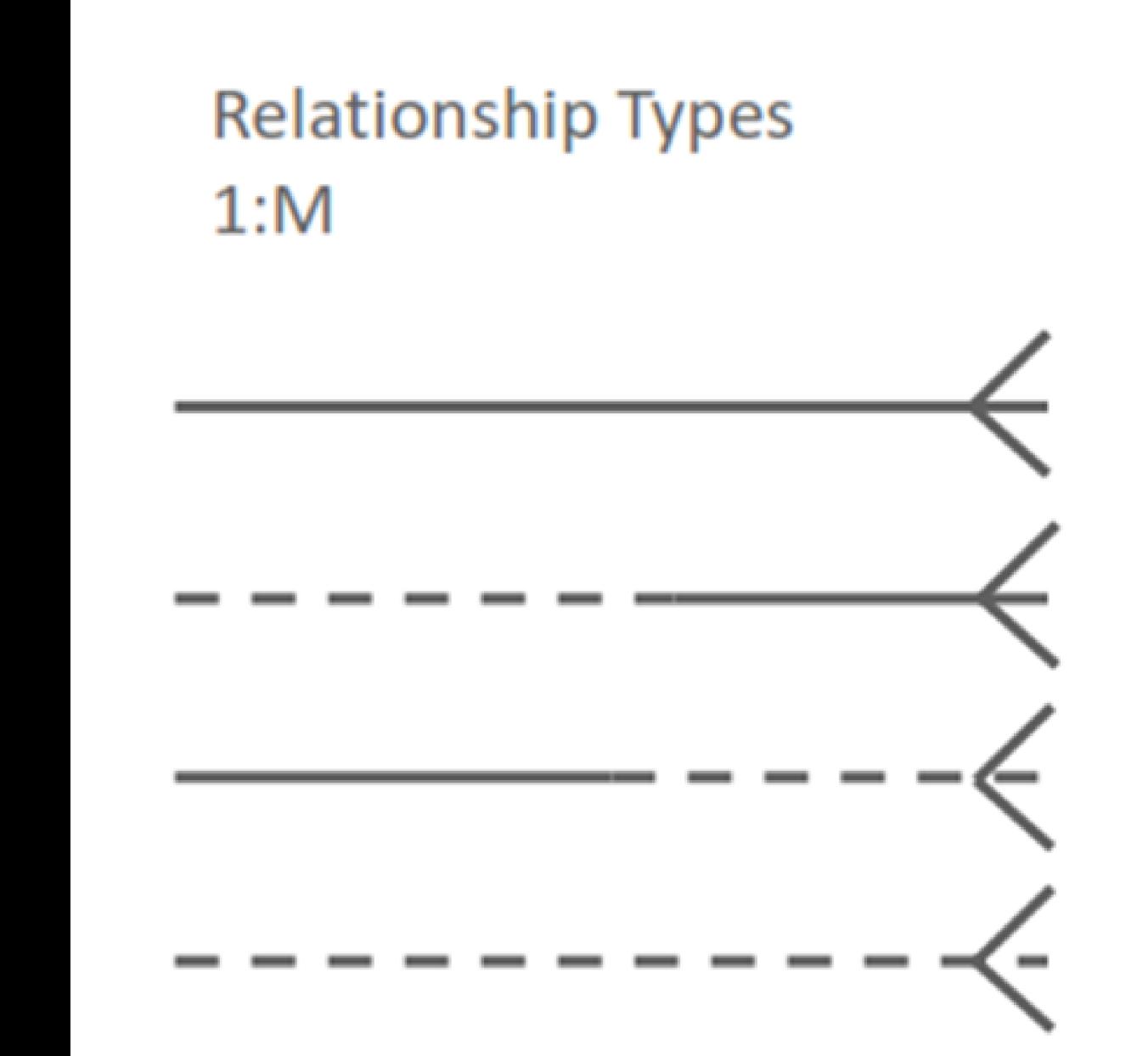
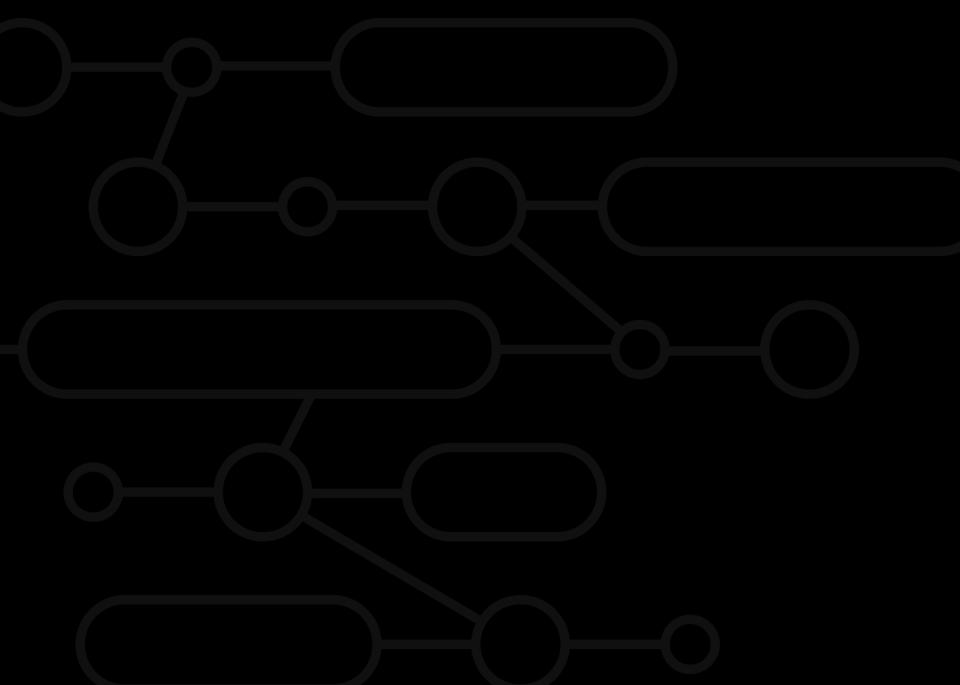
One to Many (1:M)

Relationships

- The various types of 1:M relationships are most common in an ER Model.
- “Many” can mean one-or-more or zero-or-more, depending on the optionality.

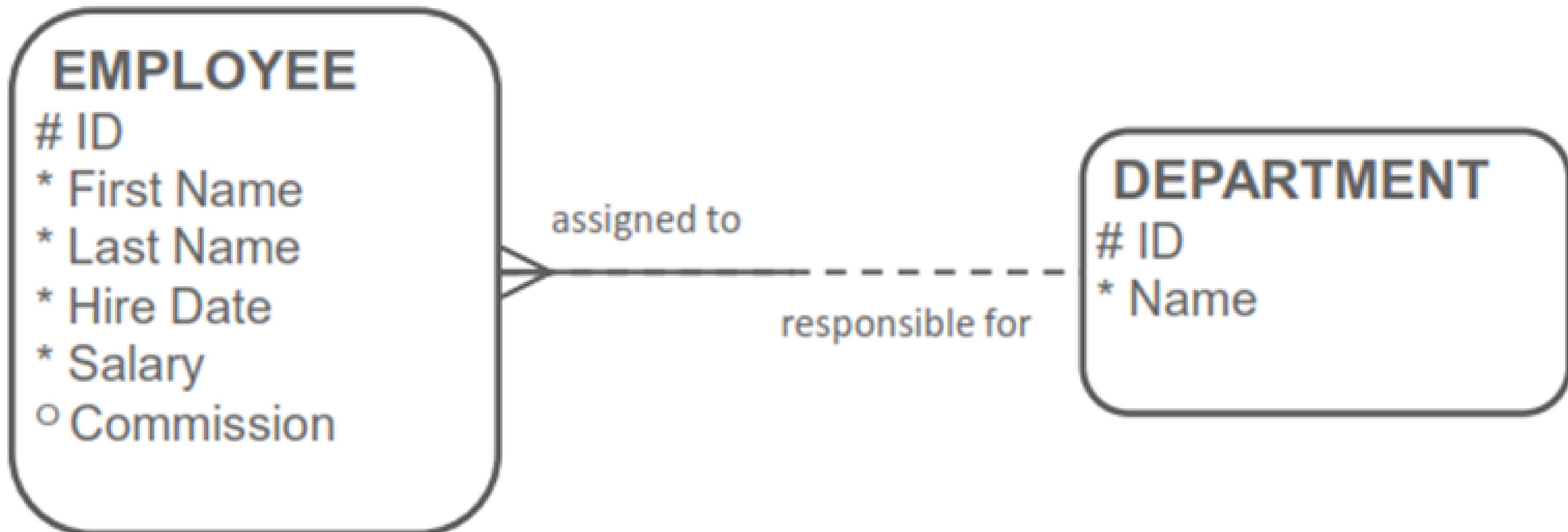
2.1 Relationship Types

One to Many (1:M) Relationships



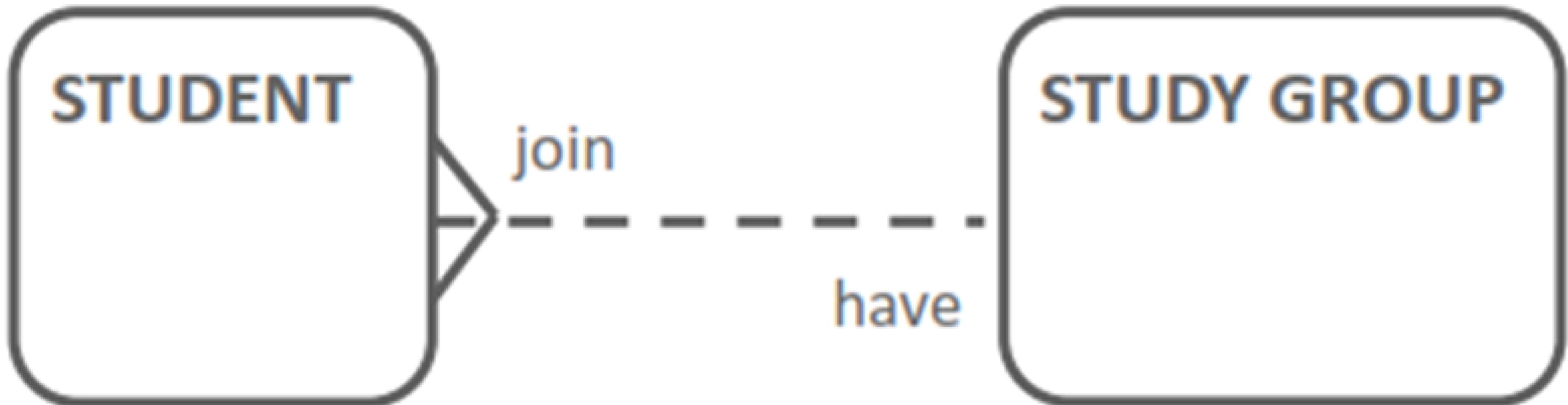
2.1 Relationship Types

One to Many (1:M) Relationships Examples



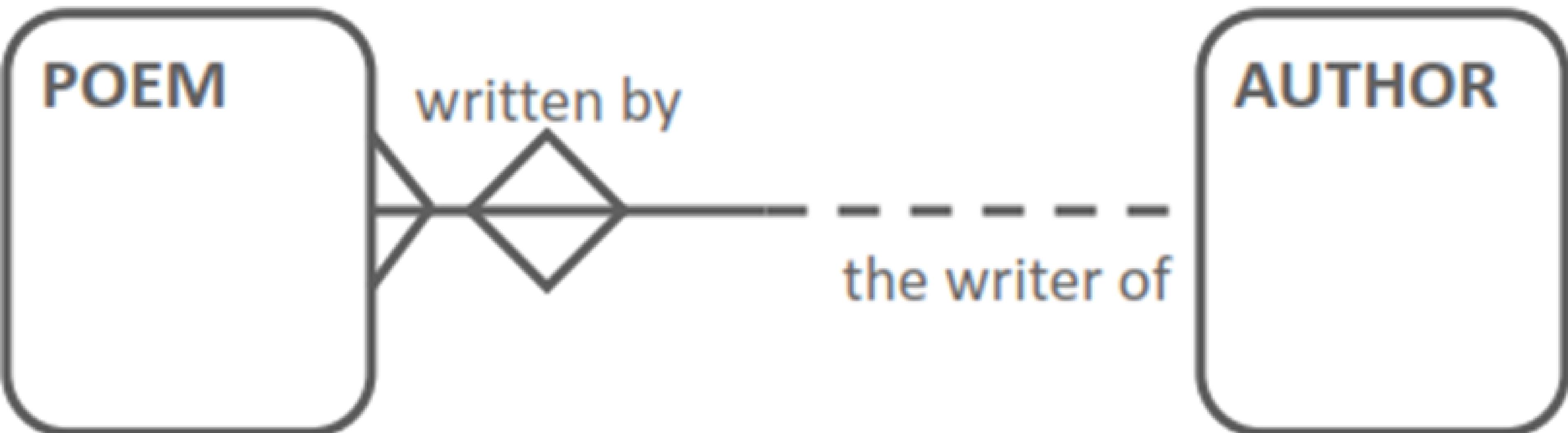
2.1 Relationship Types

One to Many (1:M) Relationships Examples



2.1 Relationship Types

One to Many (1:M) Relationships Examples



2.1 Relationship Types

Many-to-Many (M:M) Relationships

- The various types of M:M relationships are common, particularly in a first version of an ER model.

2.1 Relationship Types

Many-to-Many (M:M)

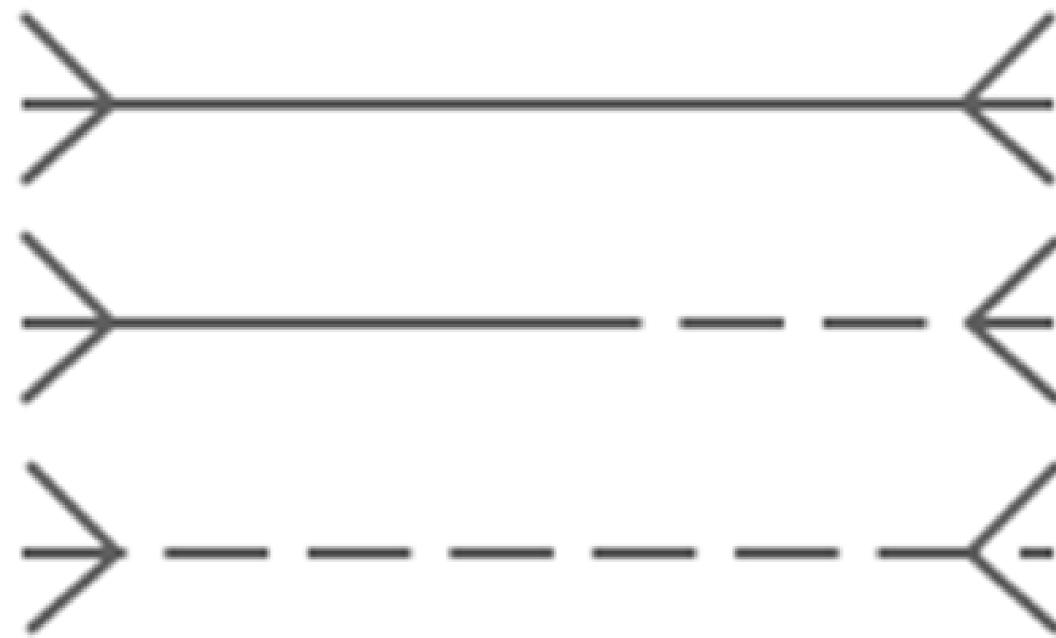
Relationships

- In later stages of the modeling process, all M:M relationships will be resolved, and disappear because many-to-many relationships indicates that an entity is missing from the model.

2.1 Relationship Types

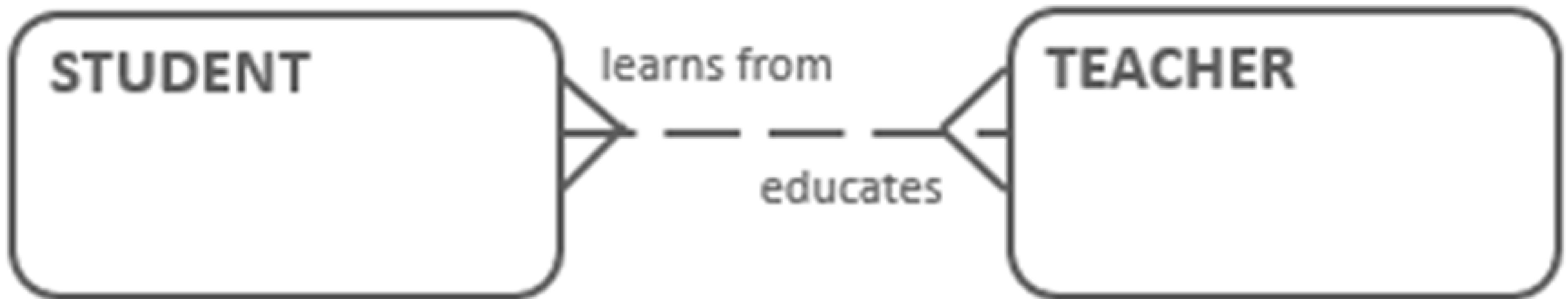
Many-to-Many (M:M) Relationships

Relationship Types
M:M



2.1 Relationship Types

Many-to-Many (M:M) Relationships Examples



M:M Relationships



One-to-One Relationships

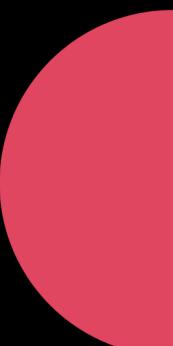
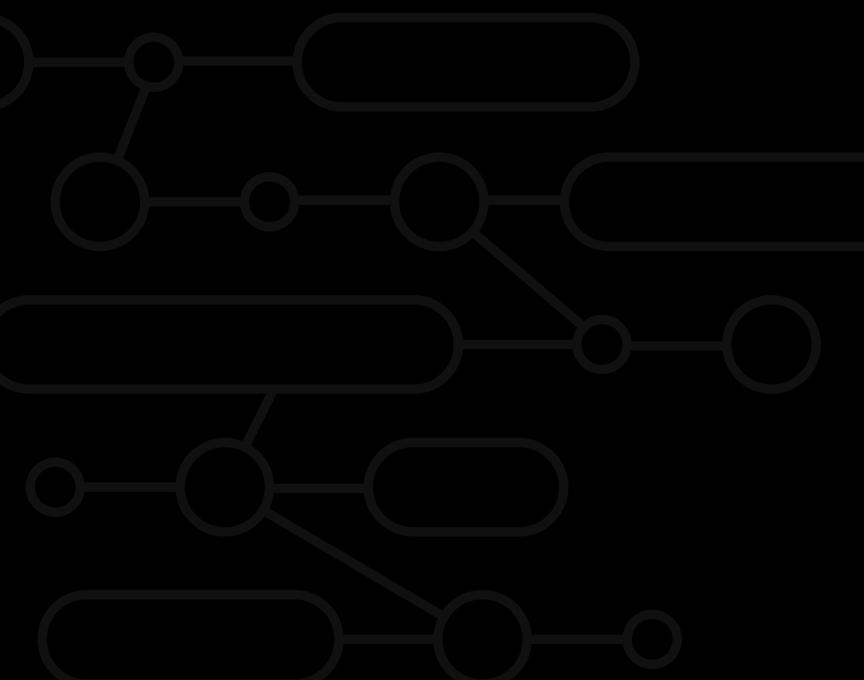
- Usually, you will find just a few of the various types of 1:1 relationships in every ER model.
- Mandatory at one end of the 1:1 relationship commonly occurs when roles are modeled.

2.1 Relationship Types

One-to-One Relationships

Relationship Types

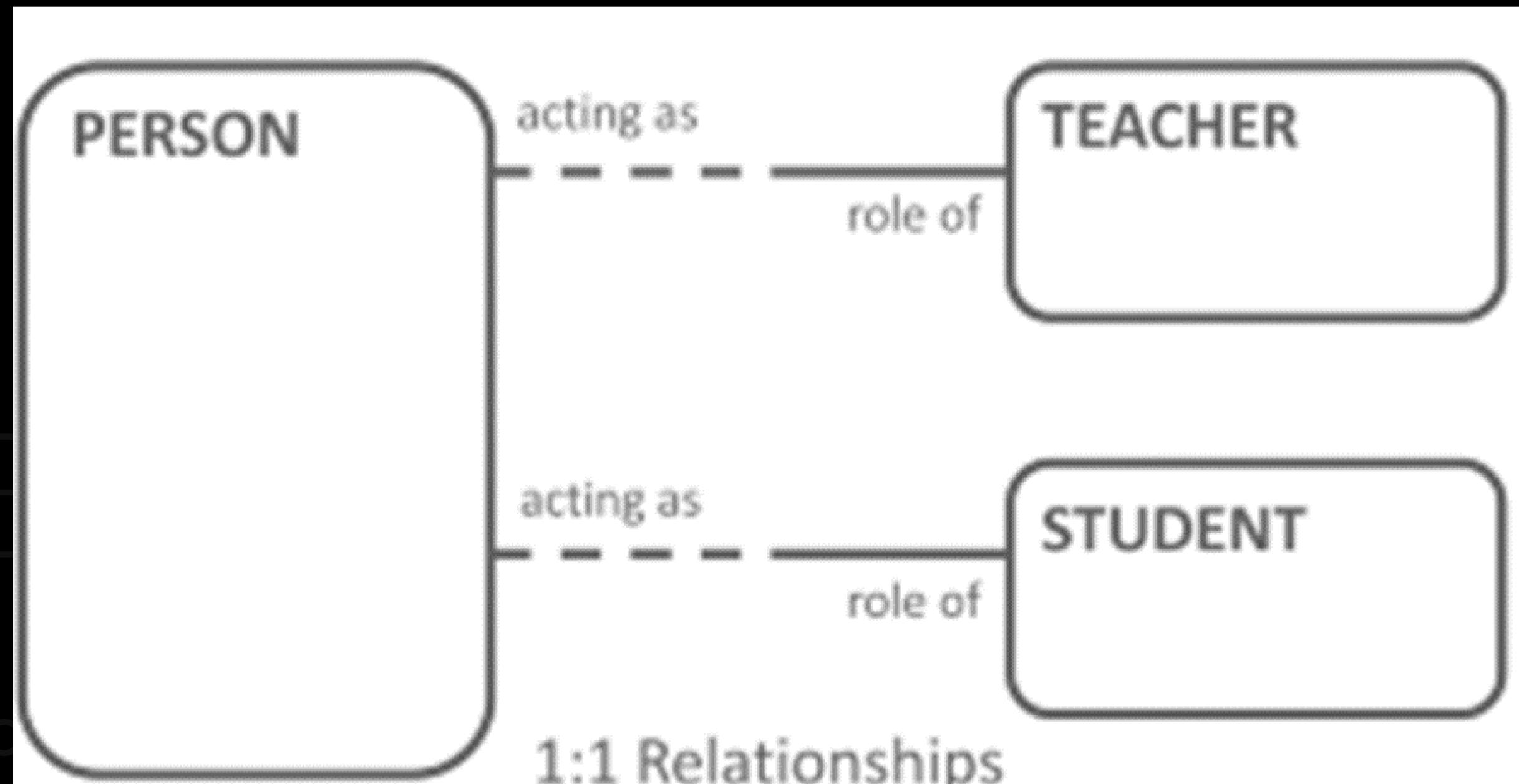
1:1



2.1 Relationship Types

One-to-One Relationships

Examples



2.1 Relationship Types

One-to-One Relationships

Examples

EXAM

evaluated with

ANSWER KEY

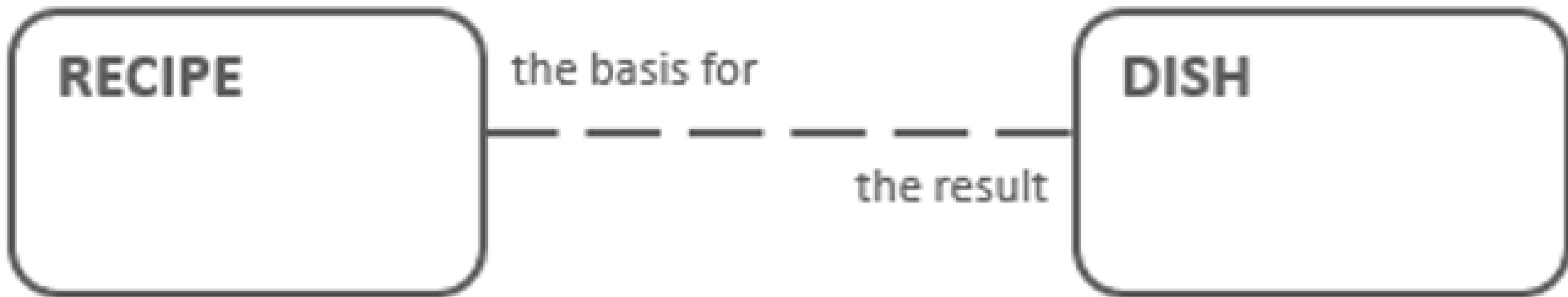
for



2.1 Relationship Types

One-to-One Relationships

Examples



1:1 Process Relationships



2.1 Relationship Types

One-to-One Relationships

- when you create an instance of one entity, there must be exactly one dedicated instance for the other simultaneously.

2.1 Relationship Types

One-to-One Relationships

- when you create an instance of one entity, there must be exactly one dedicated instance for the other simultaneously.

3.1 Redundant Relationships

Redundant Relationships

- Redundancy means something unnecessarily repetitive; the state of being unnecessarily repetitive

Redundant Relationships

- A redundant relationship can be derived from another relationship in the model, so be careful
- Redundant relationships must be eliminated to avoid confusions.

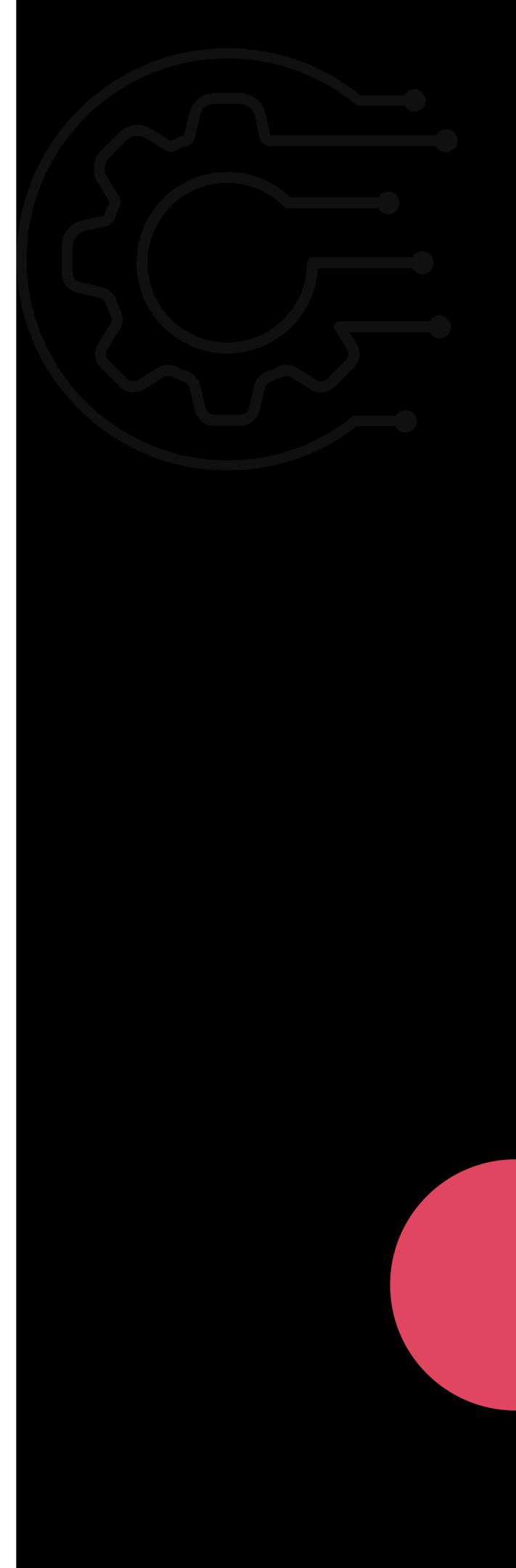
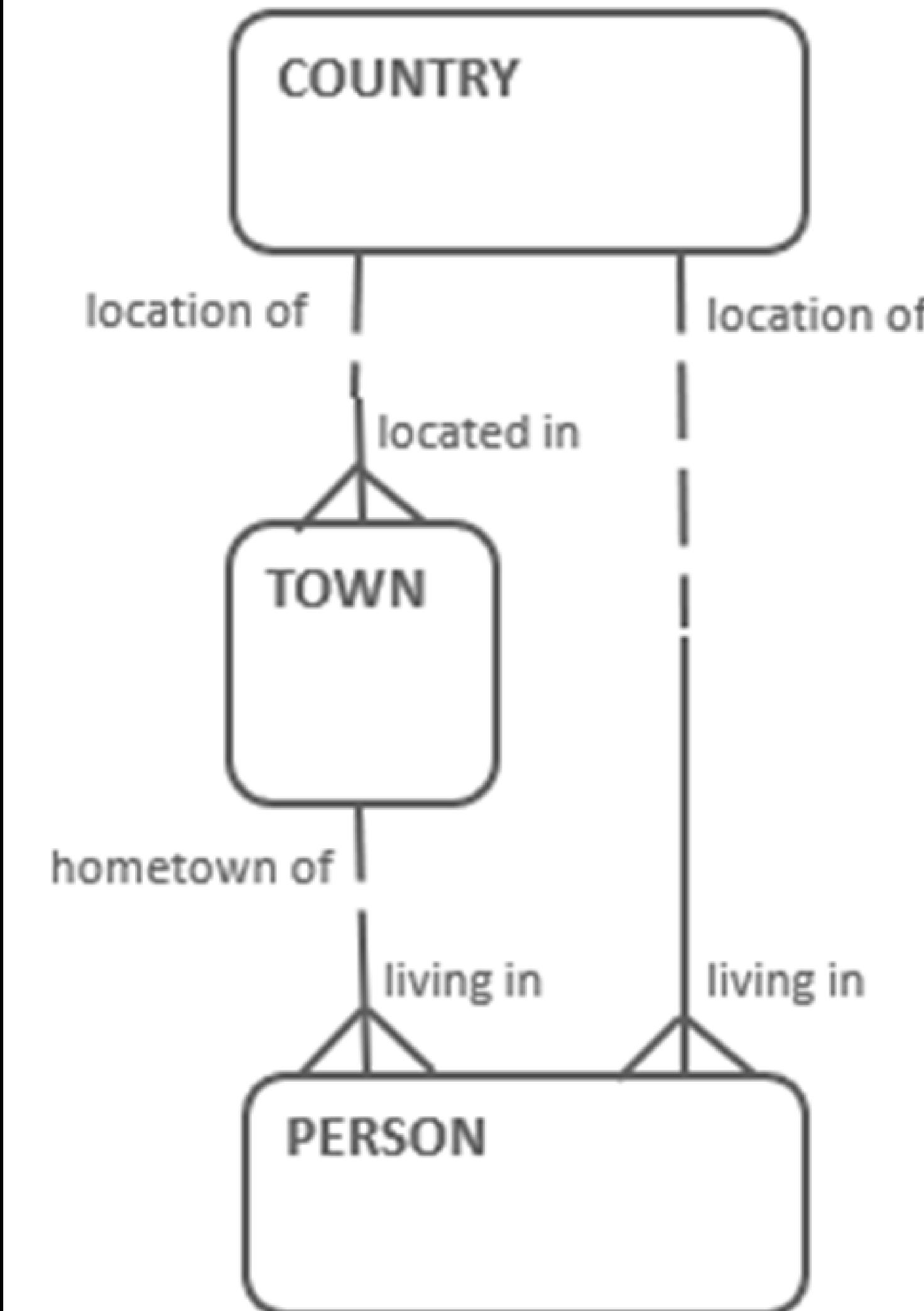
3.1 Redundant Relationships

Example:

- A database for citizenship is created.
We take note of the country they are
born in and the current place they live
at.

3.1 Redundant Relationships

Example:



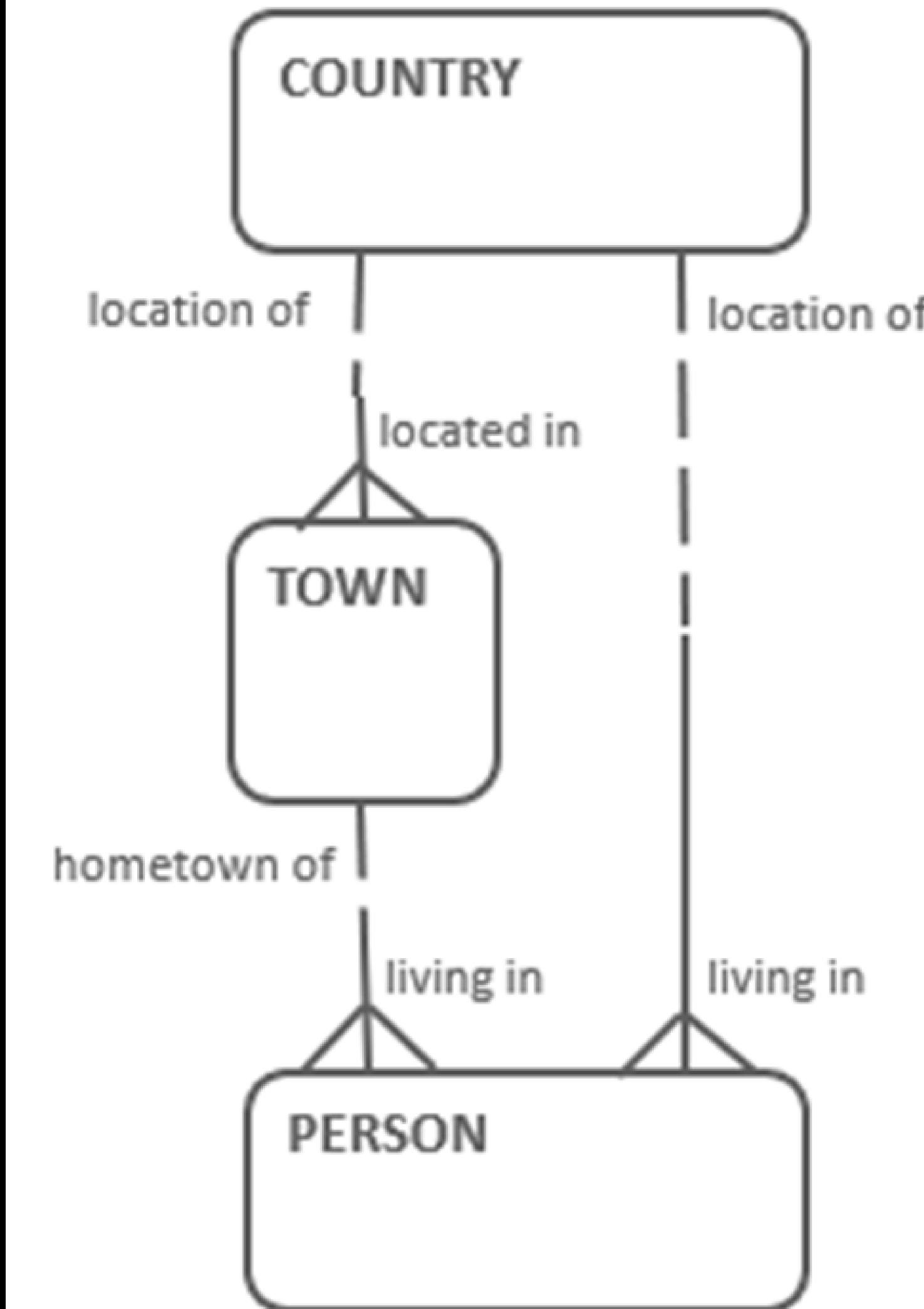
3.1 Redundant Relationships

Redundant Relationships

- In this example, you can derive the relationship from PERSON to COUNTRY from the other two relationships (COUNTRY to TOWN, TOWN to PERSON), so you should remove the direct relationship from COUNTRY to PERSON.

3.1 Redundant Relationships

Example:



3.1 Redundant Relationships

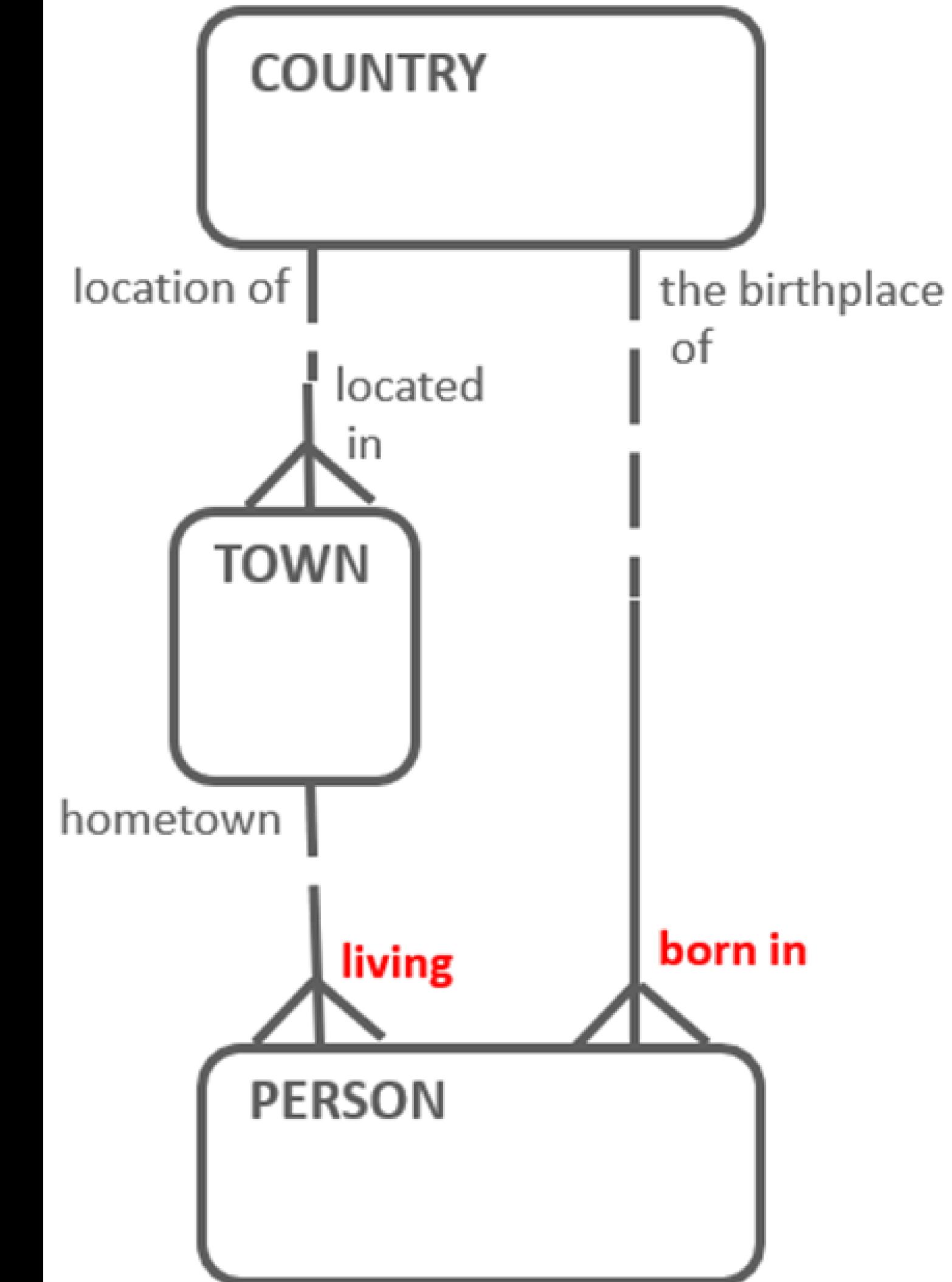
Redundant Relationships

- However, be careful of concluding that a relationship is redundant based on the structure alone.
- Read the relationships to check, and fix by removal or editing it.

3.1 Redundant Relationships

Example Resolution

- Now the new ERD does not reflect redundant relationship.



4.1 Resolving Many-to-Many Relationships

Resolving Many-to-Many Relationships

- You may not notice it but for Many - to - Many Relationships, this could lead to problems when designing your database because it could give confusion to you as the database administrator.

4.1 Resolving Many-to-Many Relationships

Many to Many Relationships

Hiding an Attribute

- some relationships may be hiding an attribute or in some cases a third entity to be created

4.1 Resolving Many-to-Many Relationships

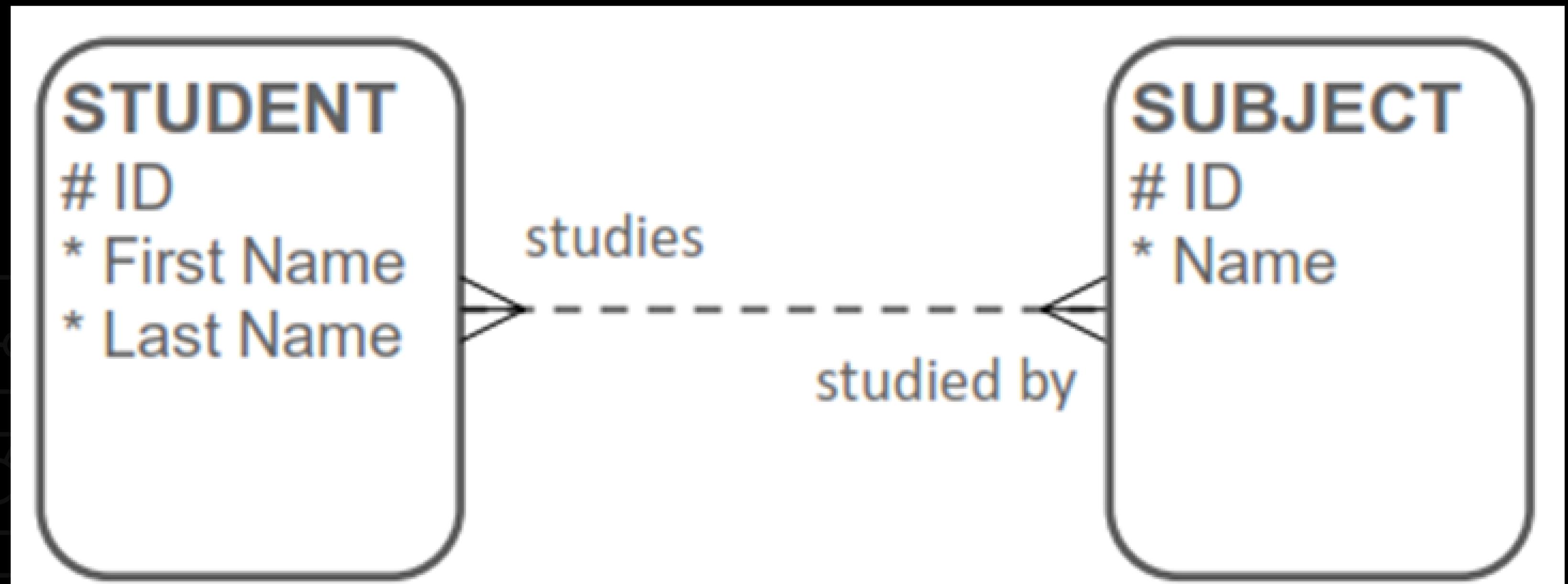
Many to Many Relationships

Hiding an Attribute Example

- In a school, a STUDENT may study one or more SUBJECTs
- Each SUBJECT may be studied by one or more STUDENTS

4.1 Resolving Many-to-Many Relationships

Many to Many Relationships Hiding an Attribute Example



4.1 Resolving Many-to-Many Relationships

Many to Many Relationships

Hiding an Attribute Example

- When a student enrolls for a subject, we want to record the grade they attain for that subject.
- Which entity would the attribute "Grade" belong to?

4.1 Resolving Many-to-Many Relationships

Many to Many Relationships

Hiding an Attribute Example

- If we put the "Grade" attribute in the STUDENT entity, how would we know which SUBJECT it is for?

4.1 Resolving Many-to-Many Relationships

Many to Many Relationships

Hiding an Attribute Example

- If we put the "Grade" attribute in the SUBJECT entity, how would we know which STUDENT got that grade?

4.1 Resolving Many-to-Many Relationships

Many to Many Relationships

Hiding an Attribute

- That's why we need to resolve the Many-to-Many Relationships.

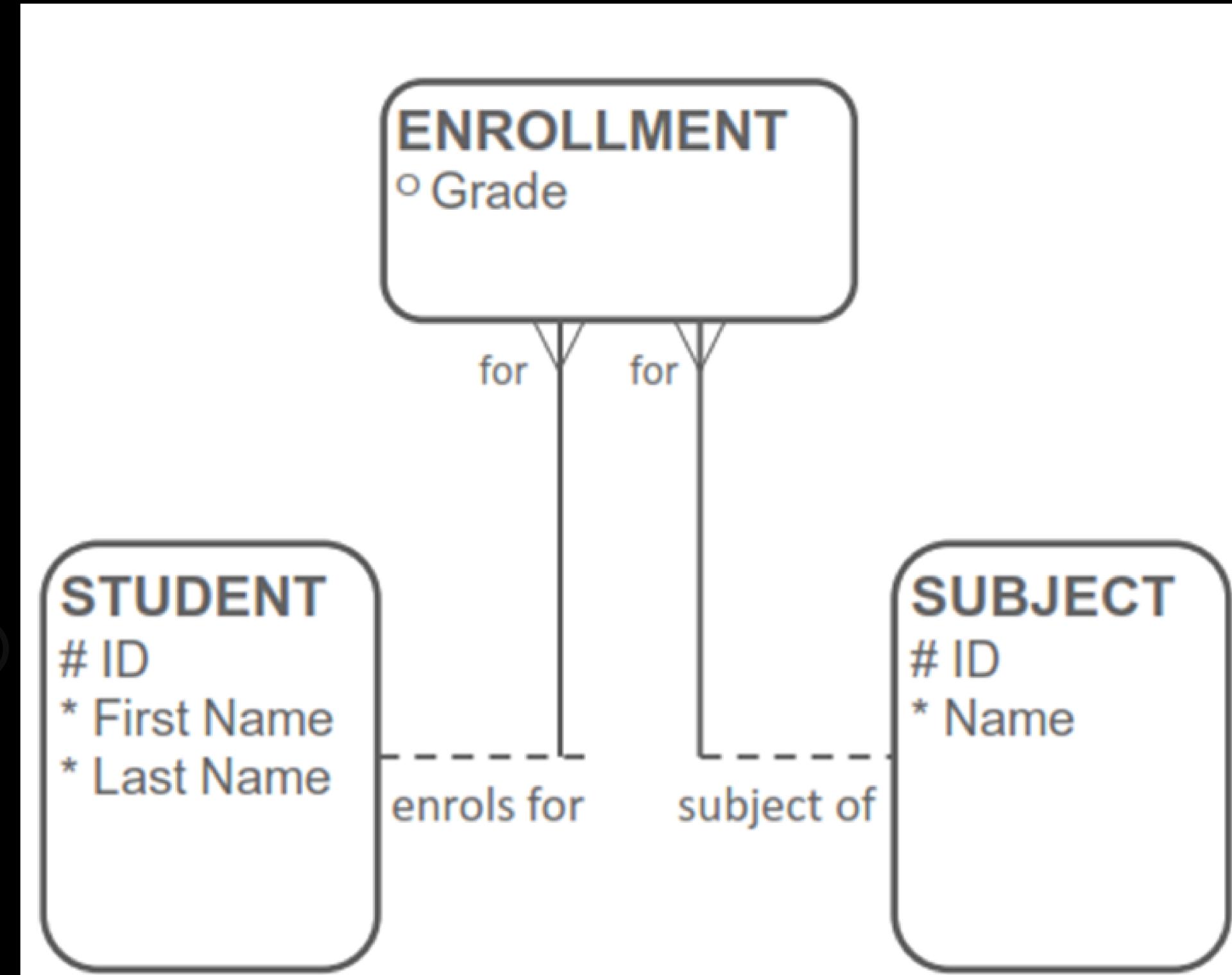
4.1 Resolving Many-to-Many Relationships

Intersection Entity

- A third entity is needed to resolve the M:M Relationship. This is called an intersection entity.

4.1 Resolving Many-to-Many Relationships

Intersection Entity



4.1 Resolving Many-to-Many Relationships

Intersection Entity

- An intersection entity

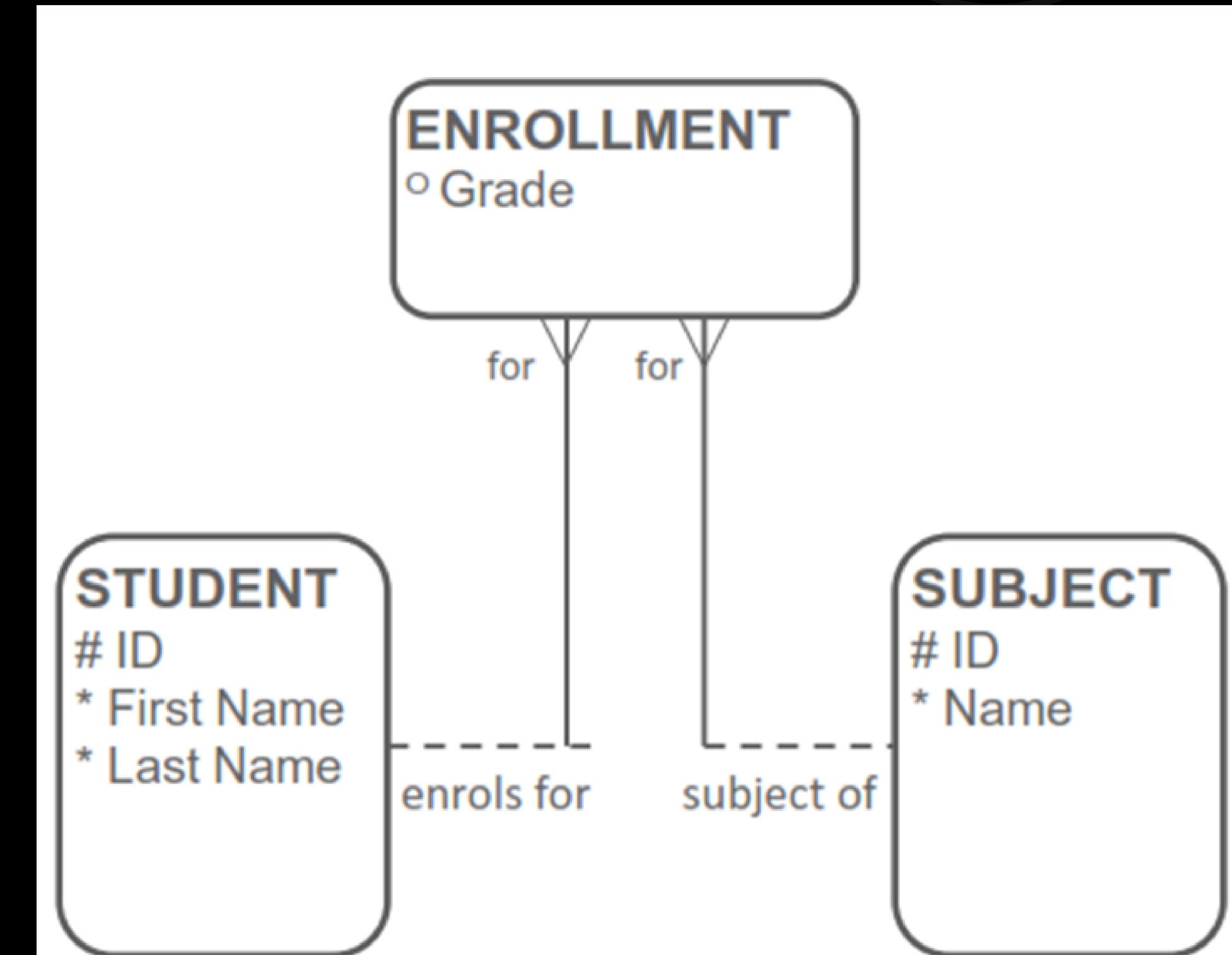
ENROLLMENT

has been added,

including the

“Grade”

attribute.



4.1 Resolving Many-to-Many Relationships

Intersection Entity

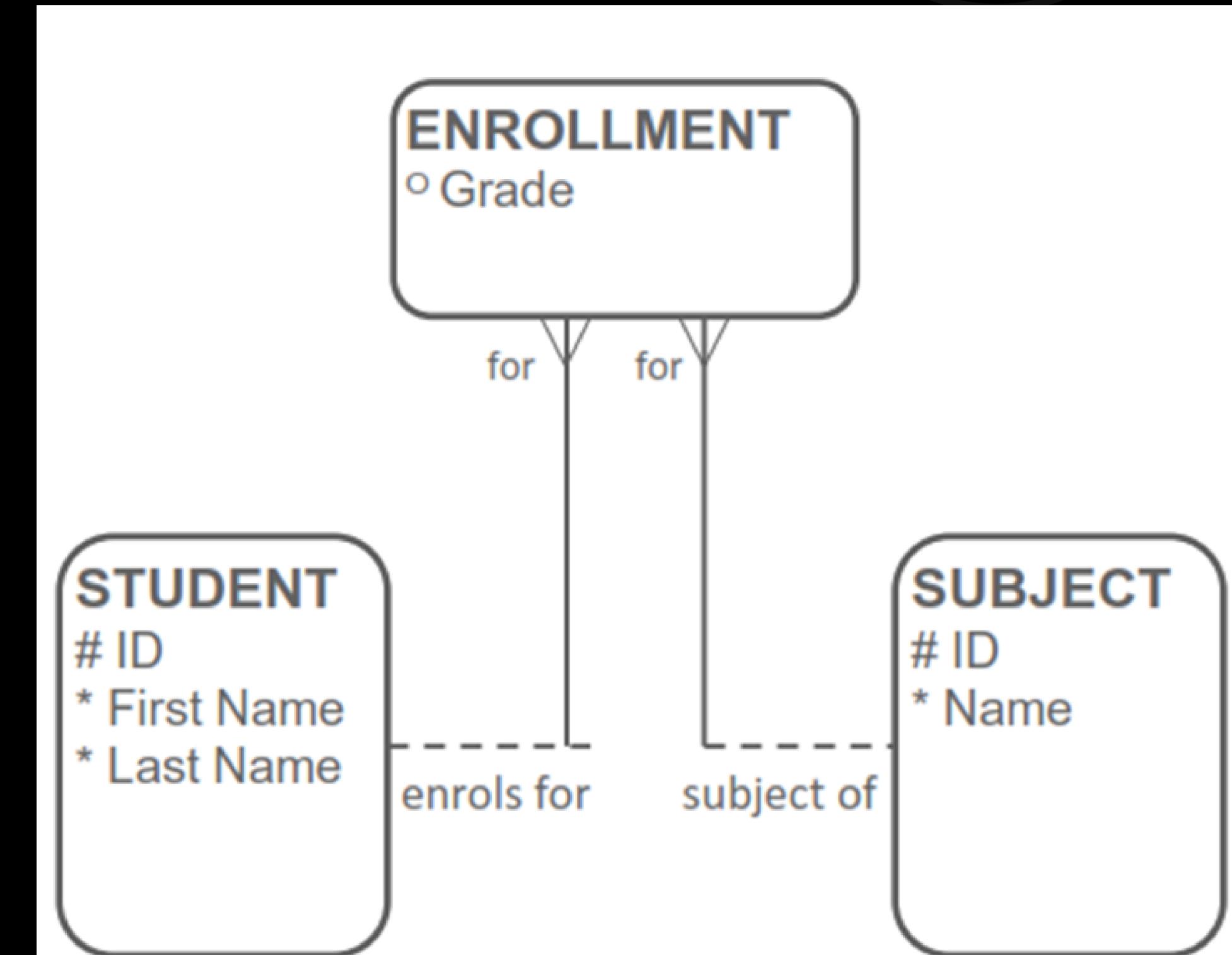
- The original

M:M

relationship has
become two

1:M

relationships.



4.1 Resolving Many-to-Many Relationships

Barred Relationships

- A relationship that participates in an entity's unique identifier
- The unique identifier (UID) of the intersection entity often comes from the originating relationships and is represented by the bars.

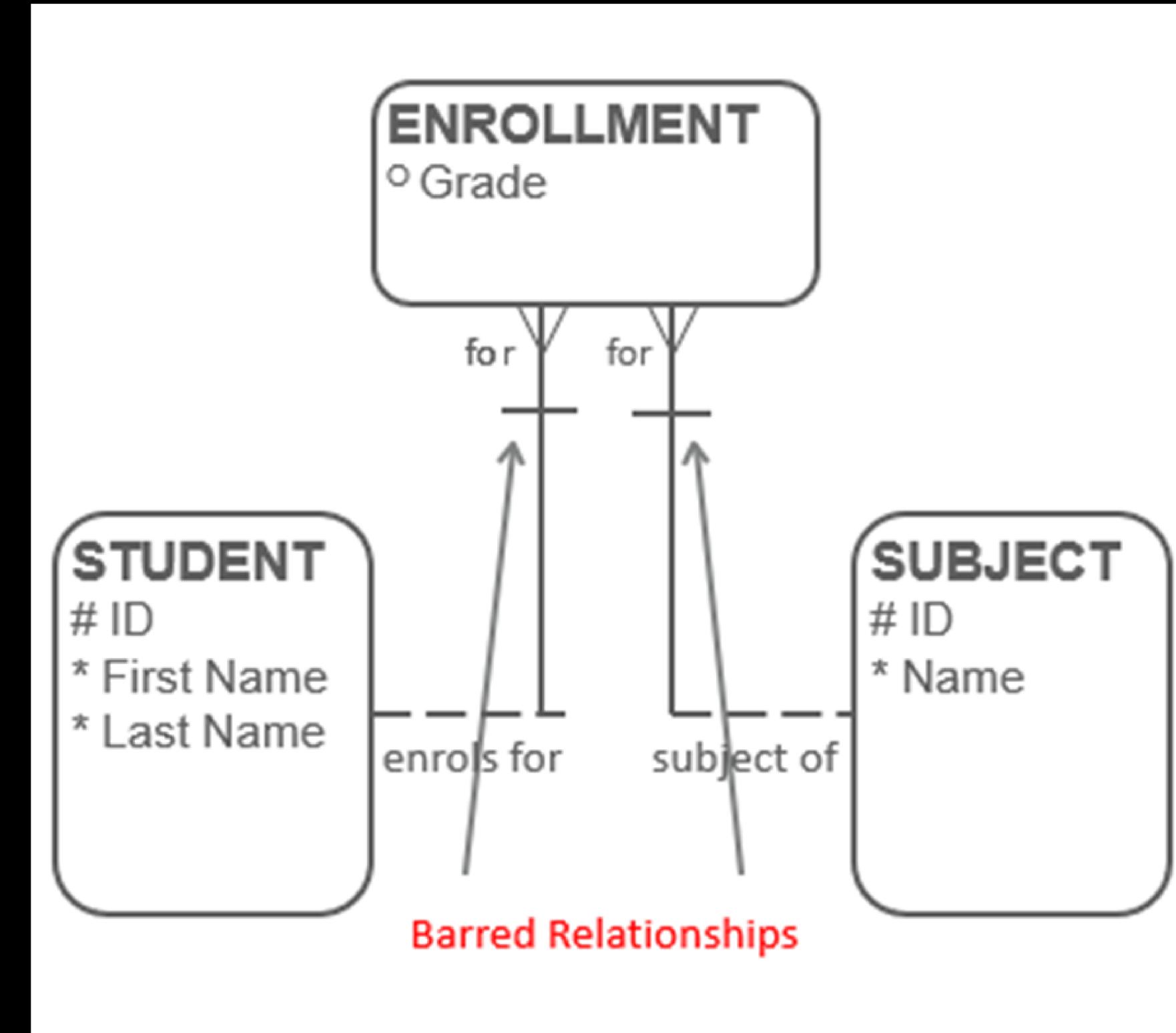
4.1 Resolving Many-to-Many Relationships

Barred Relationships

- In this case, the relationships from the originating entities to the intersection entity are called "barred" relationships

4.1 Resolving Many-to-Many Relationships

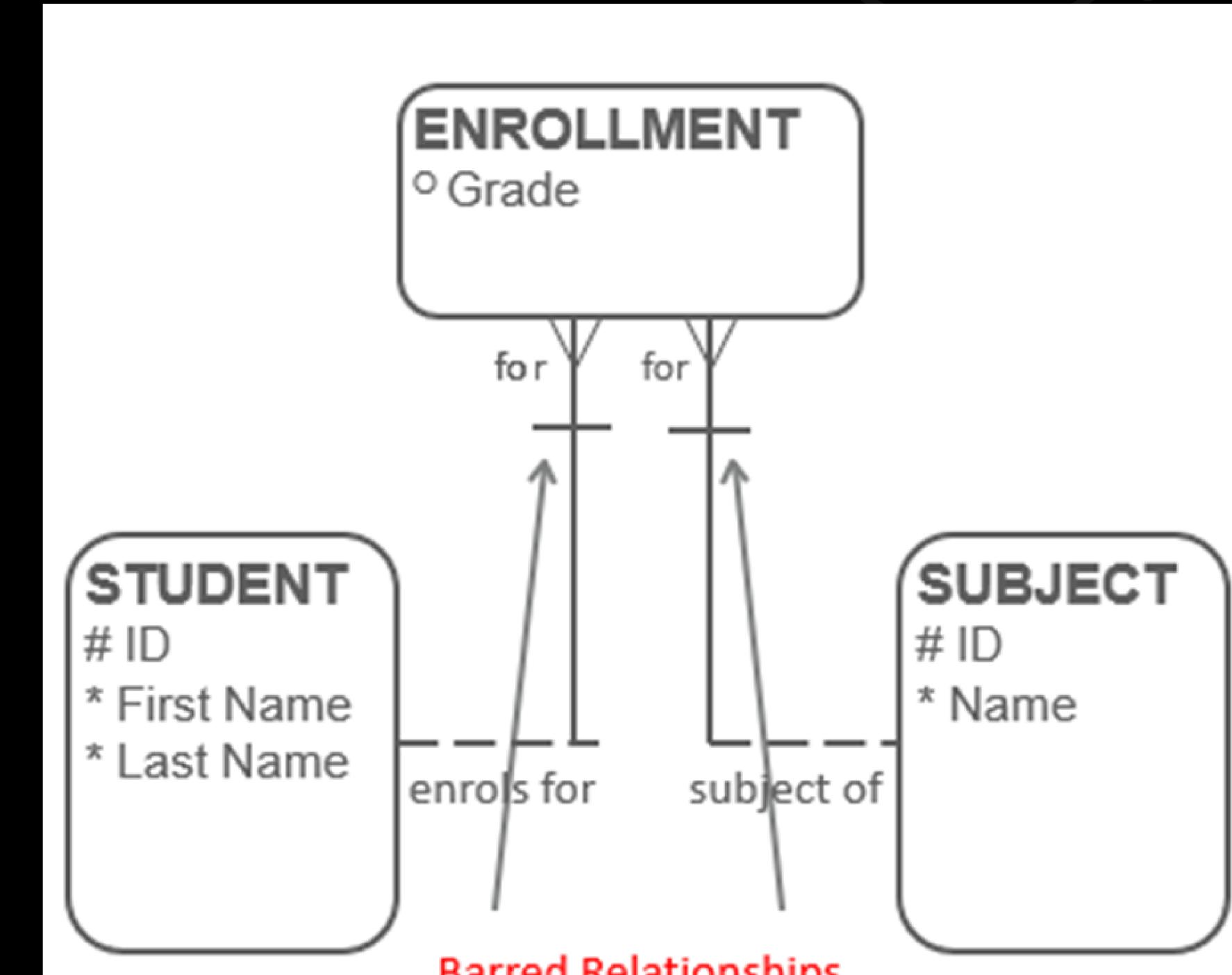
Barred Relationships Example



4.1 Resolving Many-to-Many Relationships

Barred Relationships Example

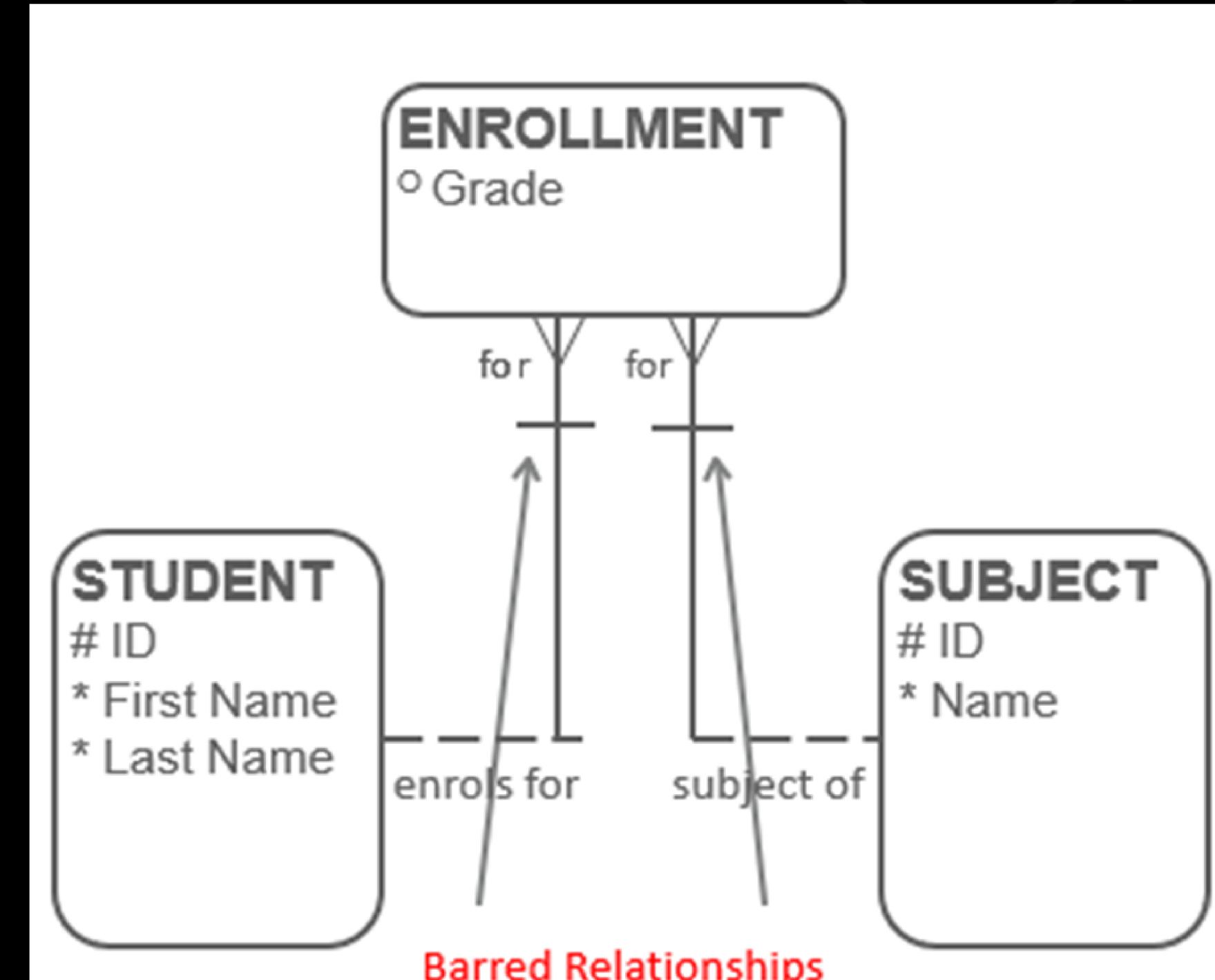
- What would be the UID of the intersection entity?



4.1 Resolving Many-to-Many Relationships

Barred Relationships Example

- The UID of the intersection entity would be coming from the originating entity, which is the student ID and the subject ID, that would be represented by bars and the relationship will be called barred relationships.



4.1 Resolving Many-to-Many Relationships

Resolving M:M Relationships Example

- Each TV SHOW may be watched by one or more PERSONS
- Each PERSON may watch one or more TV SHOWs

4.1 Resolving Many-to-Many Relationships

Resolving M:M Relationships Example



4.1 Resolving Many-to-Many Relationships

Resolving M:M Relationships Example

- If we will place attribute “Rating”, where would we place it? Is it for the PERSON entity or TV SHOW?

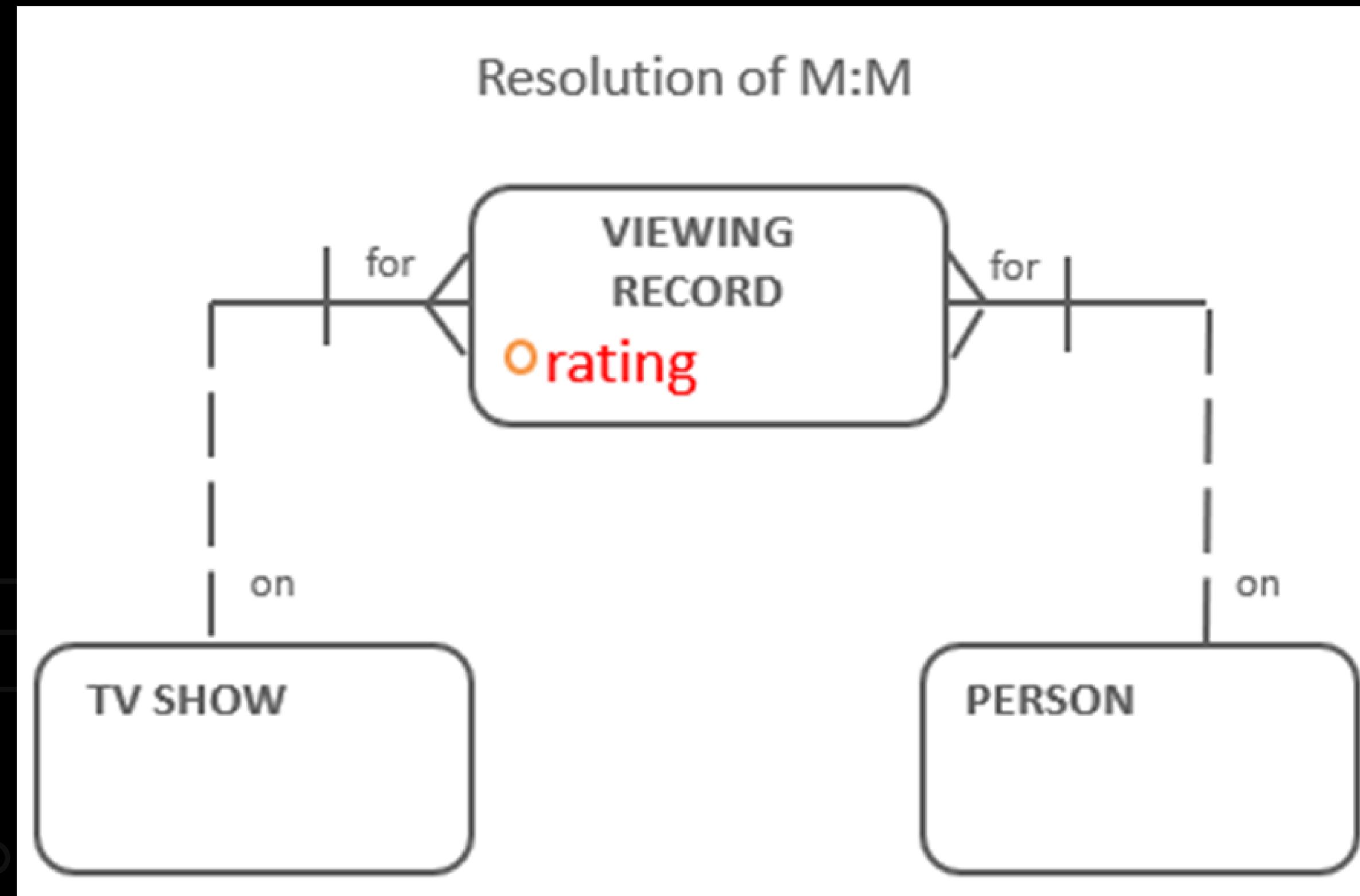
4.1 Resolving Many-to-Many Relationships

Resolving M:M Relationships Example

- If we place the attribute Rating in the TV SHOW entity, how would we know who would rate that show?
- If we place the attribute Rating in the PERSON entity, how would we know what show is the person rating?

4.1 Resolving Many-to-Many Relationships

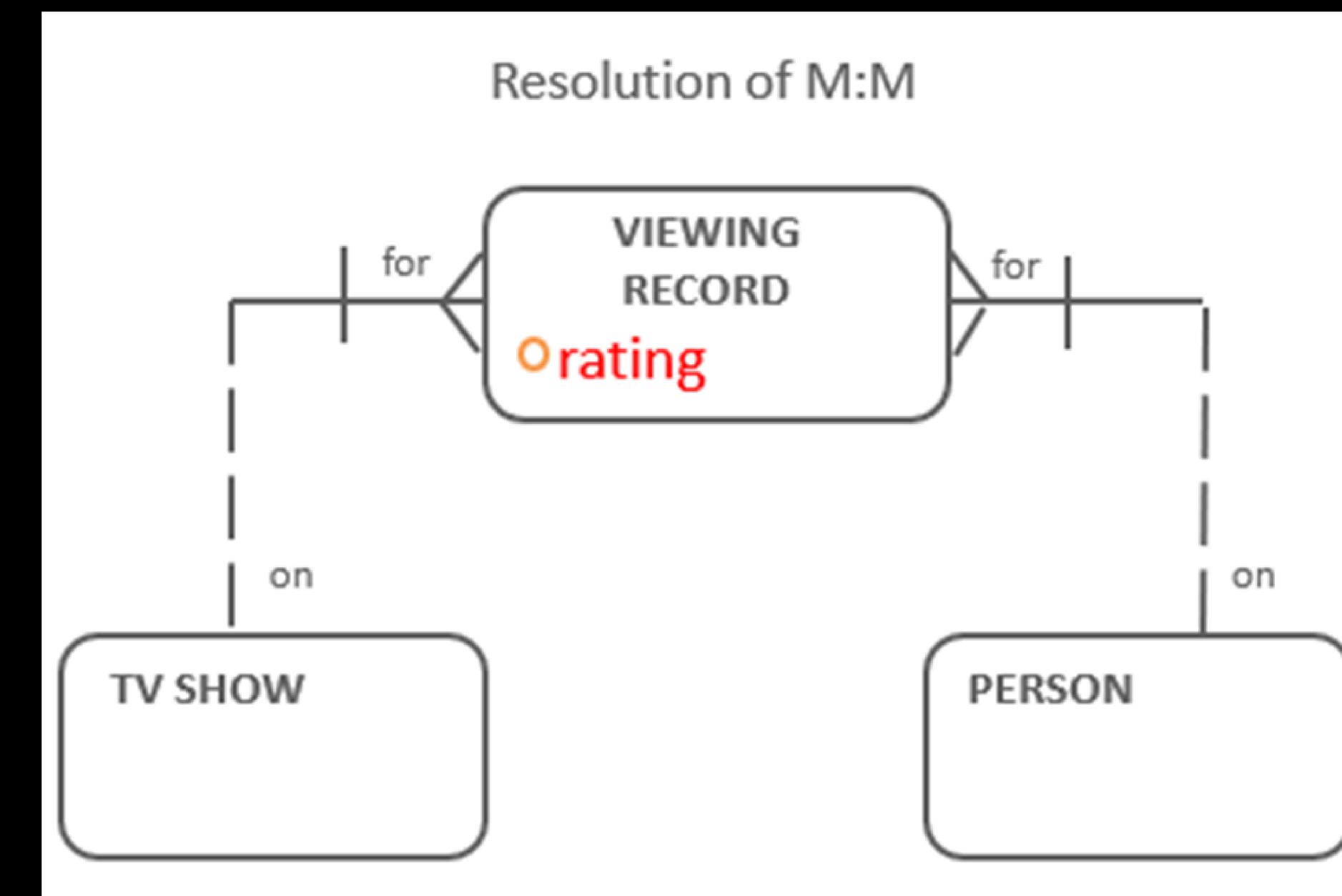
Resolving M:M Relationships Example



4.1 Resolving Many-to-Many Relationships

Resolving M:M Relationships Example

- So, we create an intersection entity **VIEWING RECORD** or **WATCHING RECORD** so that you may resolve the many to many relationship.



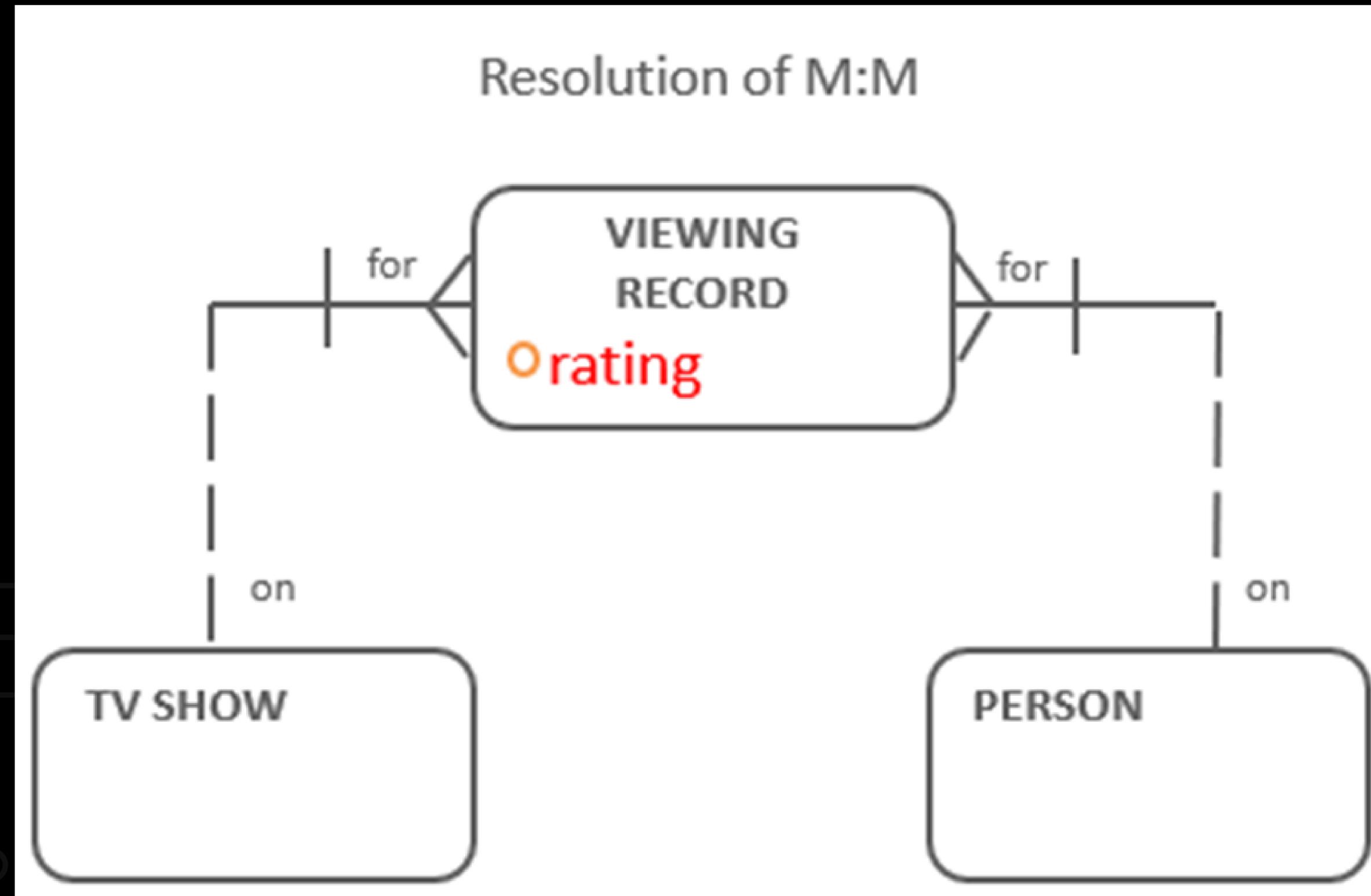
4.1 Resolving Many-to-Many Relationships

Resolving M:M Relationships Example

- Always remember the structure of how many-to-many relationships are resolved, most likely it would always have a mandatory relationship on the side where the bars are placed. This means that VIEWING RECORD could not exist without the TV SHOW and PERSON entity.

4.1 Resolving Many-to-Many Relationships

Resolving M:M Relationships Example



5.1 CRUD Analysis

CRUD Analysis

- CRUD is an acronym for create, retrieve, update, delete.

5.1 CRUD Analysis

CRUD Analysis

- These are the important keywords that you might want to look for when you are given a scenario or during interview of a client for you to be able to model correctly and accurately the important information of a business.

5.1 CRUD Analysis

CREATE Function

- During the client interview, and while writing the business scenarios and rules, look for keywords like: INPUT, ENTER, LOAD, IMPORT, RECORD, & CREATE
 - These all indicate that a record is created in the database at this time.

5.1 CRUD Analysis

CREATE Function Example

- "Whenever we get a new customer, we take down basic information (name, address, email) and assign an ID."
(CREATE)

5.1 CRUD Analysis

RETRIEVE Function

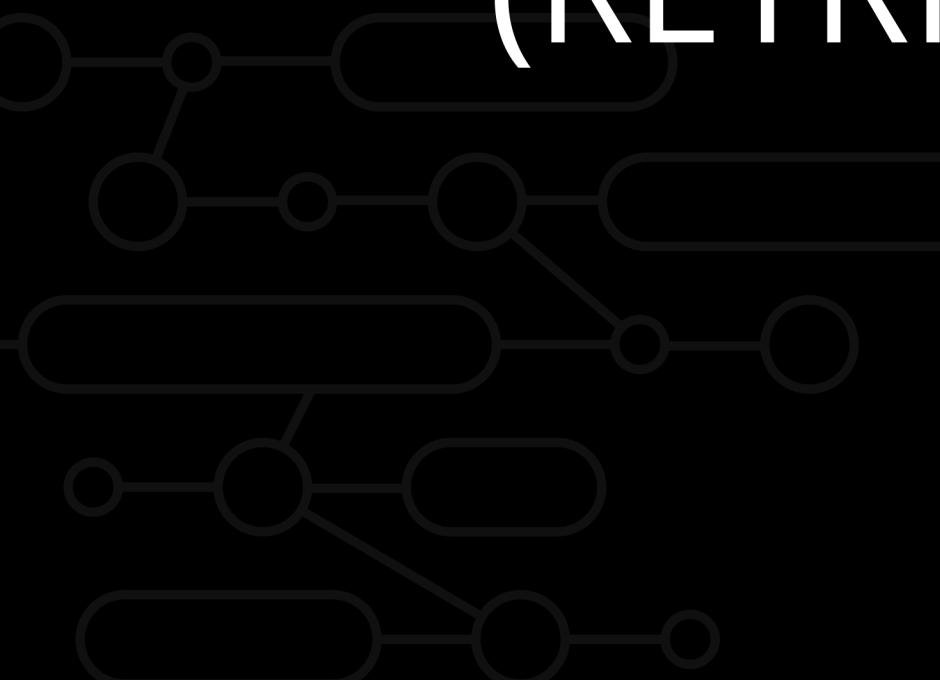
- During the client interview, and while writing the business scenarios and rules, look for keywords like: VIEW, REPORT, BRING UP, PRINT, FIND, READ, & LOOK UP
 - These all point to retrieving information from the database.

5.1 CRUD Analysis

RETRIEVE Function Example



- “We’d like to print out a list of the customers whose account is in arrears.”
(RETRIEVE)



5.1 CRUD Analysis

UPDATE Function

- During the client interview, and while writing the business scenarios and rules, look for keywords like: CHANGE, MODIFY, ALTER, & UPDATE
- These all point to updating information that is already in the database.

5.1 CRUD Analysis

UPDATE Function Example

- “When a customer pays their account, we alter their record by modifying the outstanding balance” (UPDATE)

5.1 CRUD Analysis

DELETE Function

- During the client interview, and while writing the business scenarios and rules, look for keywords like: DISCARD, REMOVE, TRASH, PURGE, & DELETE
- These all point to deleting information that is already in the database.

5.1 CRUD Analysis

DELETE Function Example

- “A number of our customers were small businesses that were hit hard by the recession. They went out of business. We deleted them from our current records.”

(DELETE)

5.1 CRUD Analysis

CRUD Analysis

- Those are the four important functions in validating requirements needed using CRUD Analysis. You are gathering important information and modeling them in the future, the things that are not important does not need to be included in your database design.