

Weak entity

- **weak entity** – an entity that cannot be uniquely identified by its attributes alone. The existence of a weak entity is dependent upon another entity called the owner entity. The weak entity's identifier is a combination of the identifier of the owner entity and the partial key of the weak entity.



WEAK ENTITY

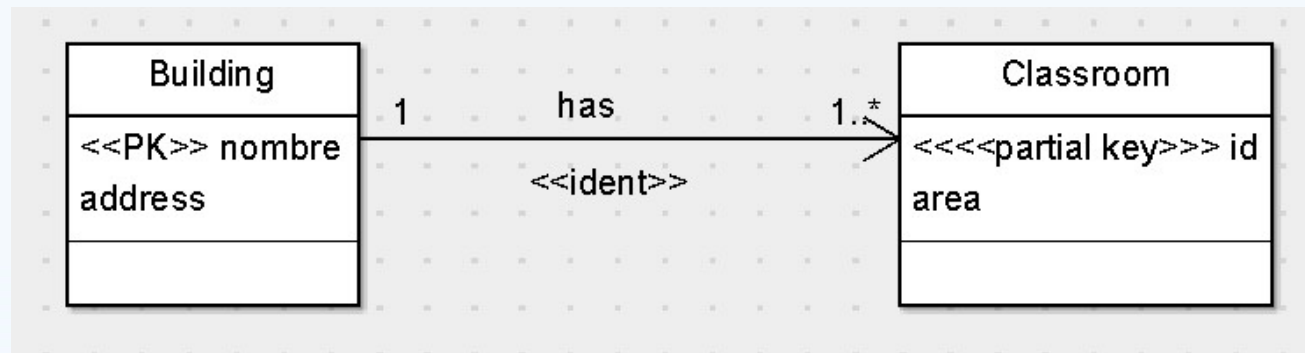
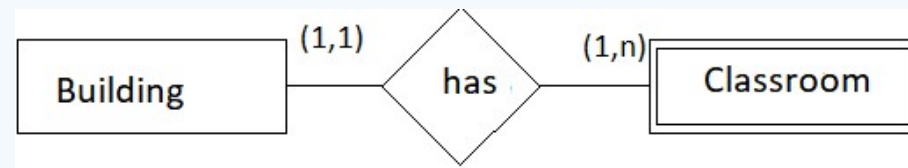
Weat Entity - example

- Consider the building and classroom entities and suppose that there may be classrooms with the same number in different buildings. Class B1.1 and B2.1 can be found in building A, and class B1.1 and B1.2 can be found in building C.

In this case, the classroom identifier does not completely identify a classroom. In order to fully identify a classroom, it is necessary to take into account in which building it is located. In fact, we can identify a classroom through the interrelation situation, which associates it with a single building. The identifier of the building where it is located together with the classroom number identifies it completely

Partial key

- *partial key attribute (discriminator)* – an attribute that, when combined with the key attribute of the owner entity, provides a unique identification for the weak entity. We underline the discriminator with a dashed line:



UML does not have special notation for weak entities

We will use the stereotypes << partial key>> to indicate the partial key and << ident >> for the identifying the strong entity.

Other authors represent weak entities without a primary key

Considerations

Sometimes, when we represent the information using the E-R model, we can find situations in which the relationships established between the entities do not collect all the requirements.

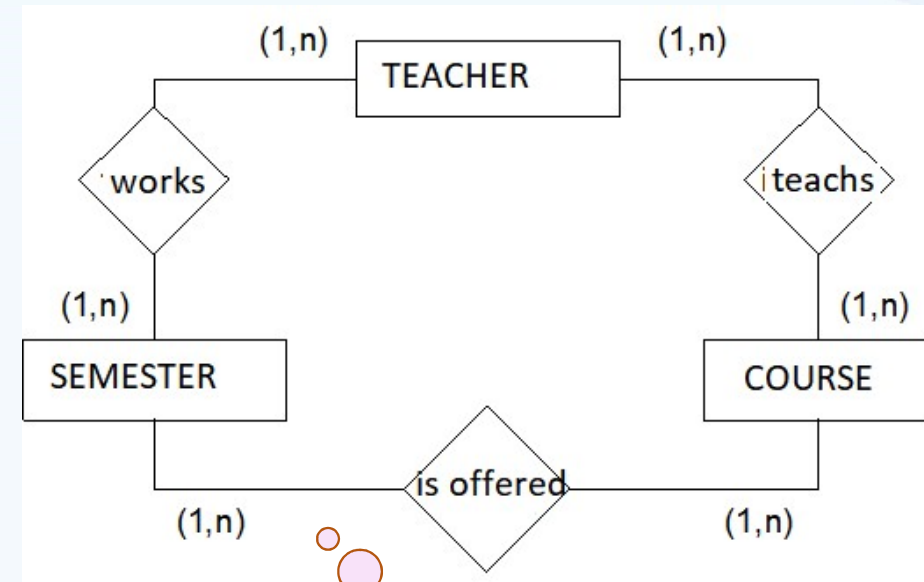
Example: you want to model the information about an academy where the teachers give courses distributed in semesters, with the following restrictions:

A teacher gives, in general, several courses and a course can be taught by several teachers

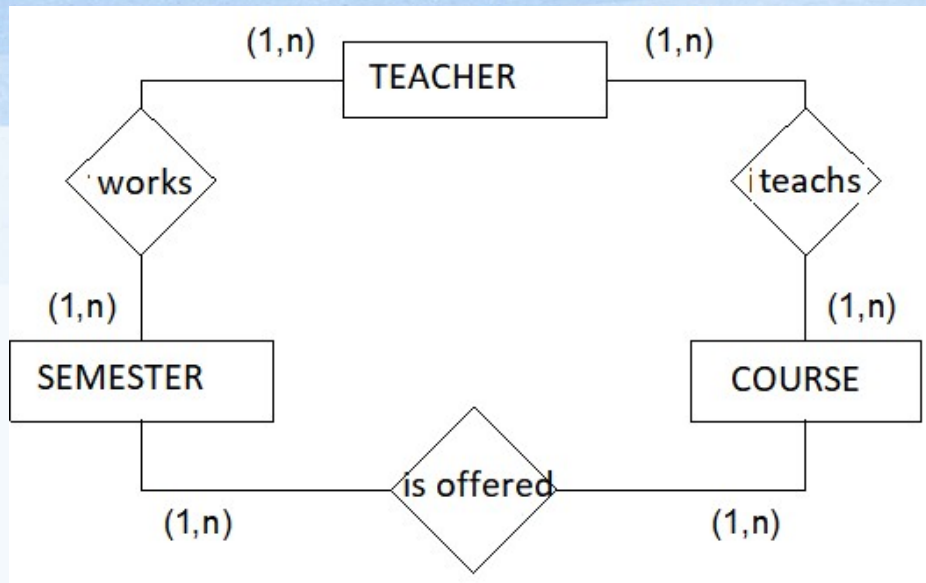
The courses can be held in different semesters, with several courses being held each semester. A course can be taught in several semesters.

It is also desired to know in what semester a teacher is working, taking into account that a teacher can teach courses in different semesters and works every semester

In general, we want to know the information of the professors with respect to the courses they teach in each term: Teacher-Course-Semester



Is this model correct?



This information doesn't imply that teacher1 works in semester2 teaching the course3 because teacher1 could teach course3 in other semester

Teacher	Semester	Course
Teacher1	Semester2	Course3

works

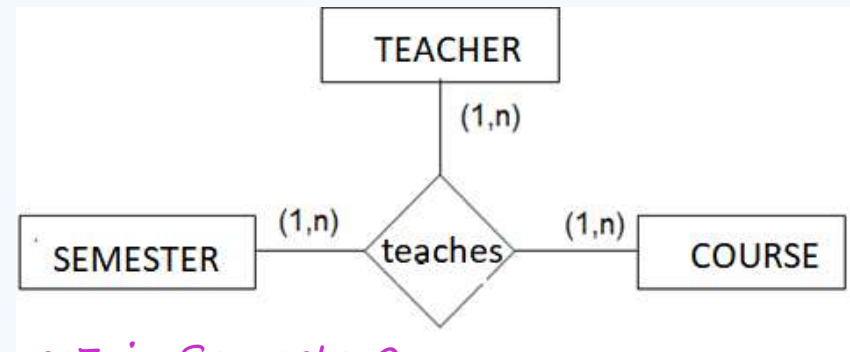
teacher	semester
teacher1	Semester2

teaches

teacher	course
teacher1	course3

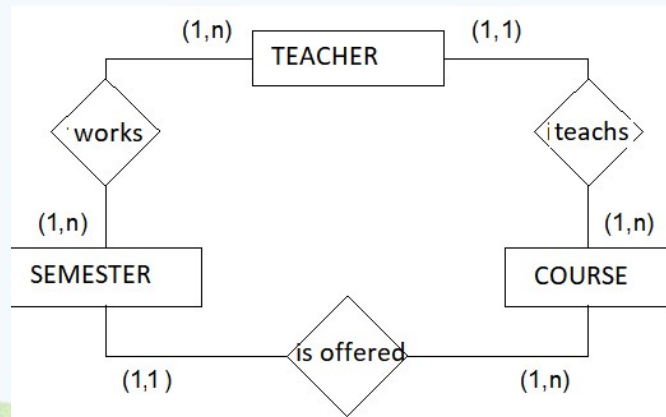
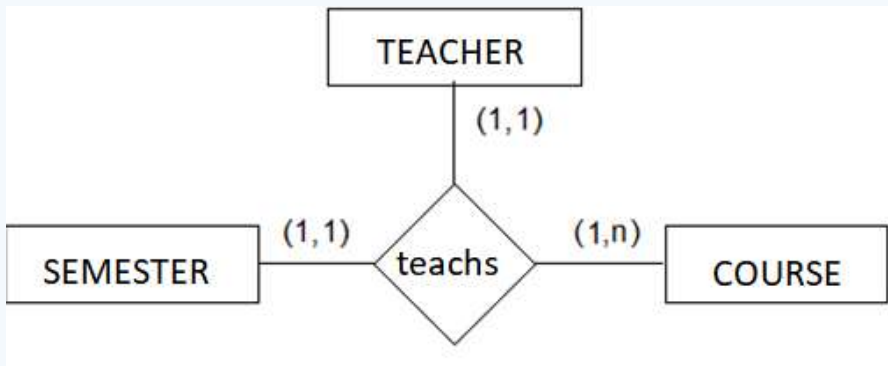
Is_offered

semester	course
semester2	course3



In this way, we can be sure that teacher1 teaches Course3 in Semester2

- Imagine same academy but this new requirements:
 - A course is only given by a teacher and is only taught in a semester
 - A teacher in a semester can teach several courses



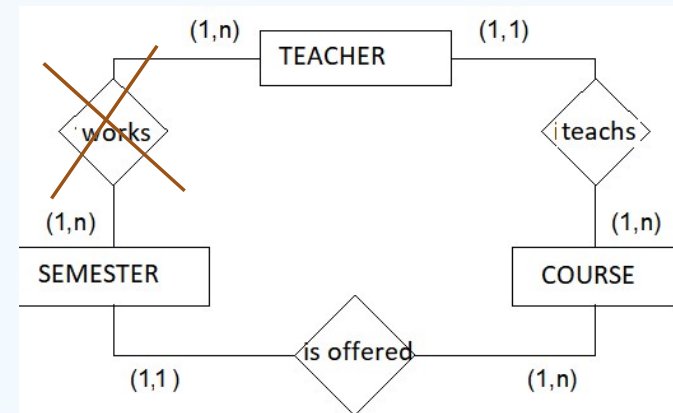
This ternary relationship, unlike the previous one, can be decomposed into three binary relations without losing meaning. This decomposition should be done (when possible) since the fewer entities participate in a relationship, the easier it will be to control the integrity of the data, and also, as we will see in the next point, in some cases it will be possible to eliminate some relationships from the scheme without loss meaning

Redundancy

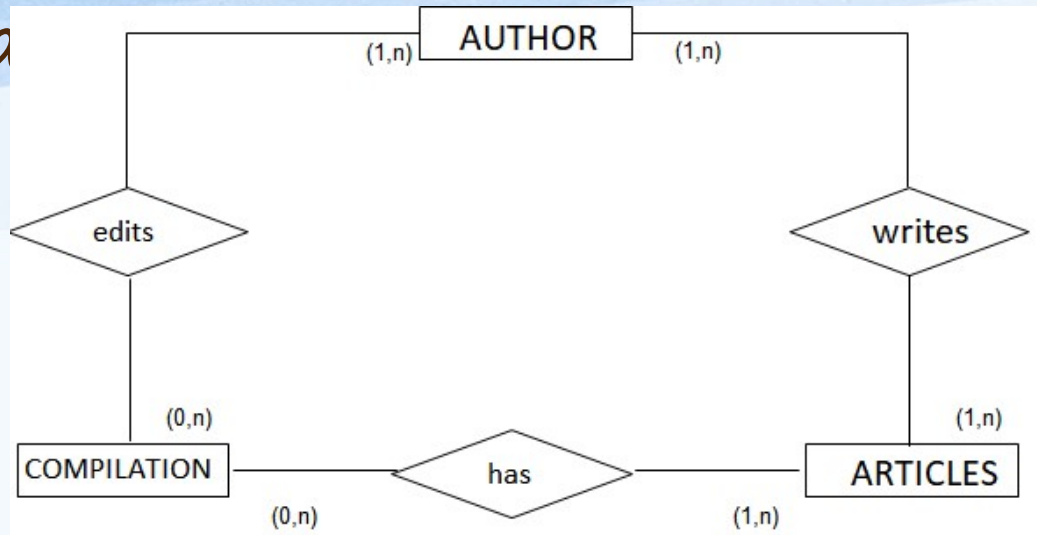
- *Data redundancy is a condition created within a database or data storage technology in which the same piece of data is held in two separate places.*
- *Non controlled redundancy can create inconsistency (not desired situation)*
- *You need to be careful with:*
 - *Existence of redundant attributes that are derived from others through some calculation:*
 - *they must be deleted or marked as redundant*
 - *Existence of cycles in the E-R diagram:*
 - *this may indicate the existence of redundant relationships*

Redundancy

- If you know the courses taught by a teacher and the semester in which it is taught, you can easily deduce in what semester that teacher has worked
- In the same way, given a semester, if we know the courses that have been taught in it, we can deduce which teachers have worked in that semester
- Therefore, the WORKS relationship between the PROFESSOR and SEMESTER entities is redundant



Redundant

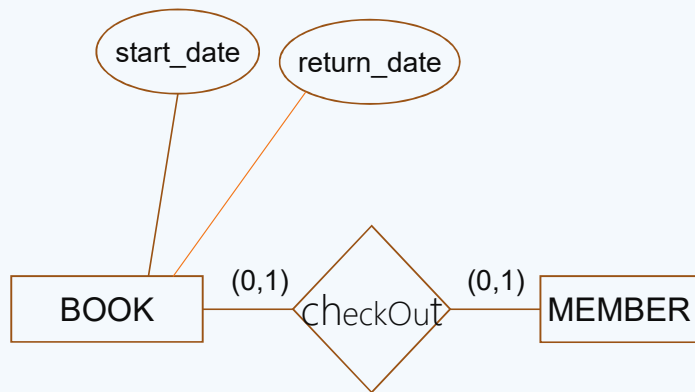


Aparece

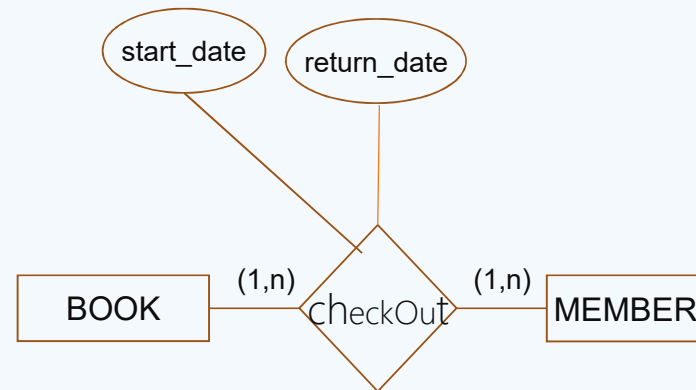
- Despite having a cycle, maybe there are no redundant relationships.
- Performing an analysis similar to the one carried out in the previous diagram, it can be verified that none of the relationships can be eliminated without loss of meaning
- This is a non-redundant cycle

Temporal Dimension of data

- When making the conceptual scheme, it is necessary to analyze whether the data that is to be stored is going to constitute a historical database or, if on the contrary, only the current state of the data is of interest.



We only need to know if the book is or is not on loan and who was the last person who rented it



In this scheme we want to save a history of who has borrowed each copy at each time, as well as the dates of start and renew