CSC 343 - Introduction to Databases

Tutorial #2:

The Entity Relationship Data Model

by Solmaz Kolahi

Brief Introduction

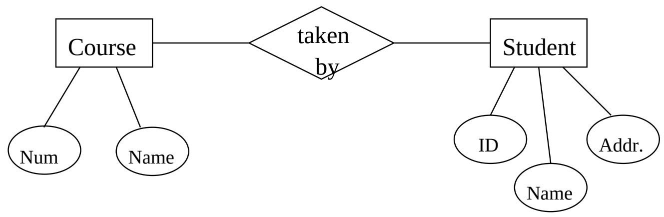
Database designers begin by developing a schema using Entity/Relationship or object-based models, and then try to convert it into relations.

Ideas $\rightarrow$ E/R Design $\rightarrow$ Relational Schema $\rightarrow$ Relational DBMS

We are focusing on the first step.
Elements of E/R model:

- Entities
- Attributes
- Relationships

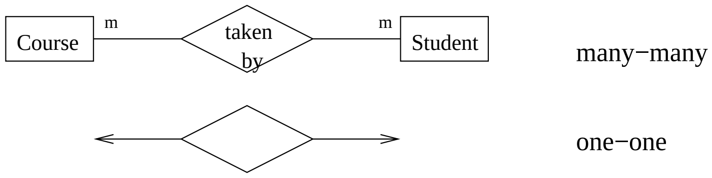
Entity Relationship Diagram is a graphical model to represent the elements. Rectangles for entities, ovals for attributes, and diamonds for relationships. A simple example:



Multiplicity

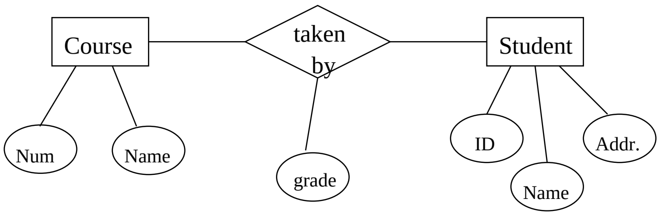
To show how many instances of one entity can associate to a single instance of the other entity.

- many-many
- many-one
- one-one



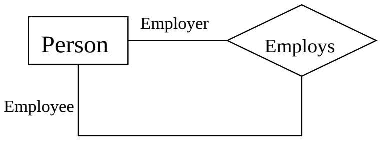
Attributes on Relationships

In the example below, *grade* is not an attribute of *Course* or *Student*, but an attribute of *taken by*.



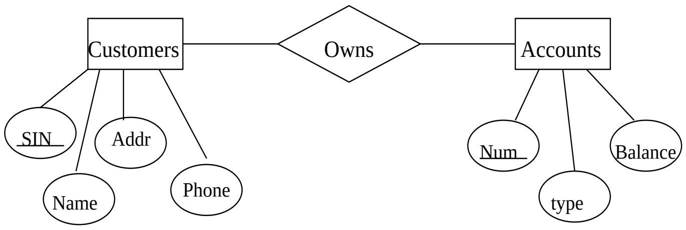
Roles

Sometimes a single entity contributes to relationships with different roles. In this case, it is essential to write roles on relationships.

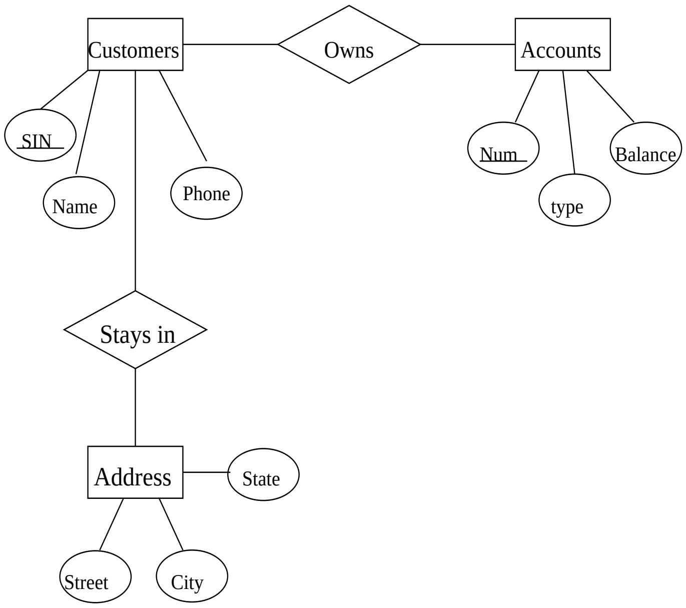


Exercise

Let us design a database for a bank, including information about customers and their accounts. Information about customers includes their name, address, phone, and SIN. Accounts have numbers, types (e.g. savings, checking) and balances. We also need to record a customer who own an account. Draw the E/R diagram for this database.

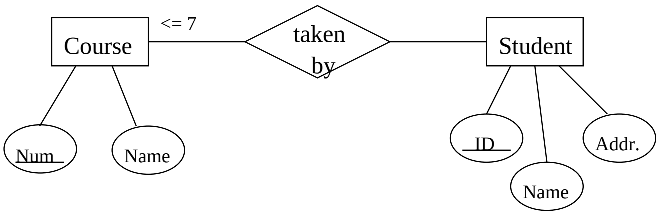


Modify the diagram so that a customer can have a set of addresses (which are street-city-state triples). Remember that we do not allow attributes to have non-atomic types such as sets.



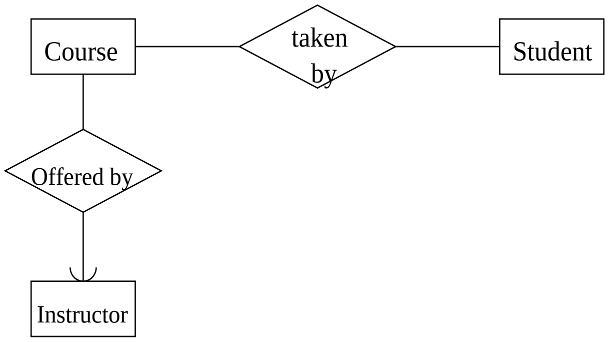
Representing Keys and Multiplicity Constraints

A key is s set of attributes for an entity, that identifies the entity uniquely; keys are underlined. In the following example, we assume that each student cannot have more than 7 courses.



Referential Integrity

A requirement that an entity be connected, through a given relationship, to an entity of some other entity set, and that the latter entity exists in the database, is called a referential integrity constraint. This ensures consistency when modifying (especially deleting) entities in a database. We use rounded arrows to show this requirement.



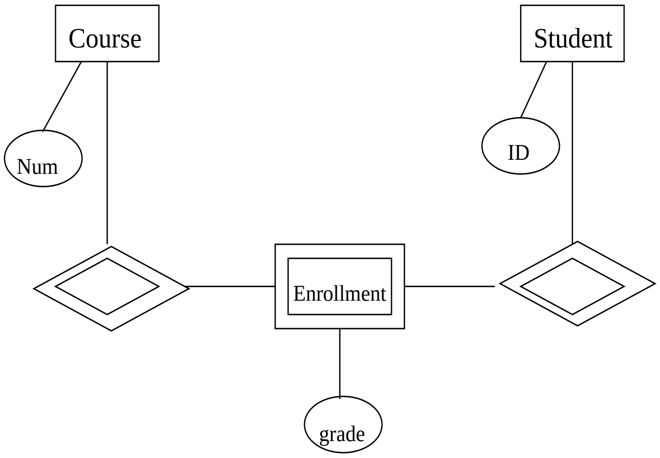
This means each course is offered by one instructor, and this instructor is present in the *Instructor* entity set. If one wants to delete an instructor, he should obey one of these rules: delete it only if the instructor does not offer any course, or delete all the courses offered by the instructor as well.

Weak Entity

An entity which does not have a unique identity by its own is called a weak entity. It will be unique in composition with the keys of other entities it relates. We use rectangles and diamonds with double borders to distinguish weak entities.

Exercise

Modify the first E/R example so that we have an entity with the name *enrollment* instead of *taken by* relationship. *Enrollment* entities form a "connecting" entity set between students and courses.



Now, *Enrollment* is a weak entity set, since each enrollment is not unique without considering the student and course associated with it.

Modify the previous diagram, so that we can record grades of the student for each of several assignments within a course.