Entity-Relationship Modelling

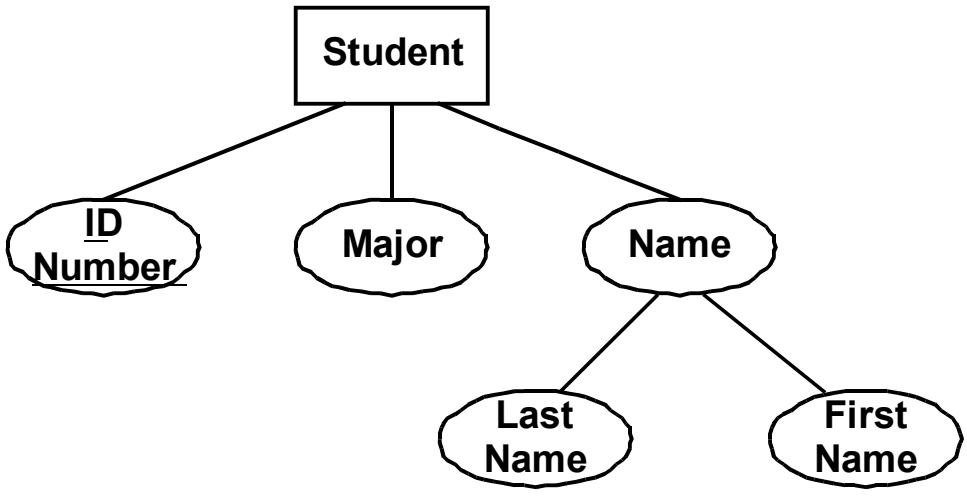
- The first step in designing a database schema is to model the underlying situation.
- This is typically carried out using a *conceptual modelling language*.
- The entity-relationship (ER) framework is one such conceptual modelling language.
- It is not used to model the schemata of database systems.
- Rather, it is used in the initial design process.
- The ER design is then translated to a relational schema.
- In the entity-relationship model, there are two basic building blocks:
    - **Entities** are “things”, often with hierarchical structure.
    - **Relationships** describe the way in which entities are related to one another.

Entities:

A simple example: In Pascal, we may define a record structure for the type student as follows.

```
Type student = record
    ID_number: ID_type;
    Name:  record
               LastName:  nametype;
               FirstName: nametype;
           End;
    Major: majortype
End;
```

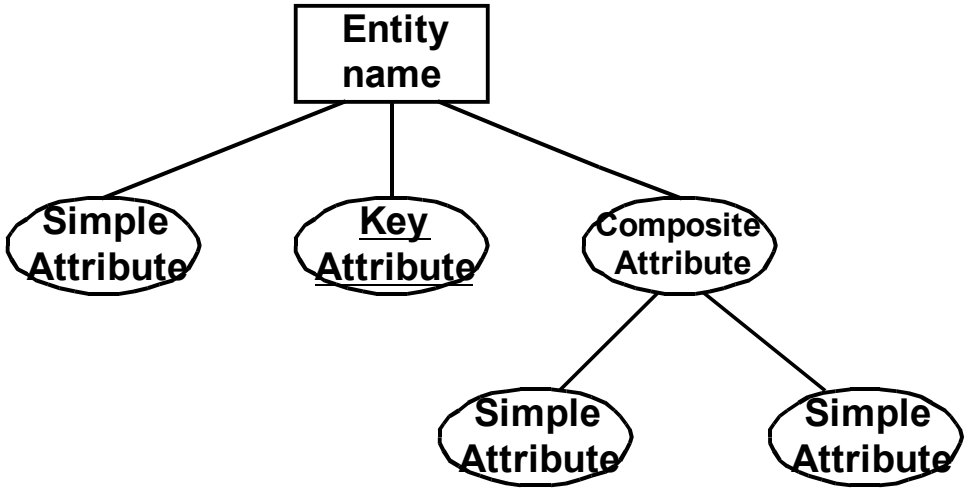
In ER modelling, such a structure may be represented diagrammatically.



Note that the types of the data items are not represented in the ER diagram.

A closer look:

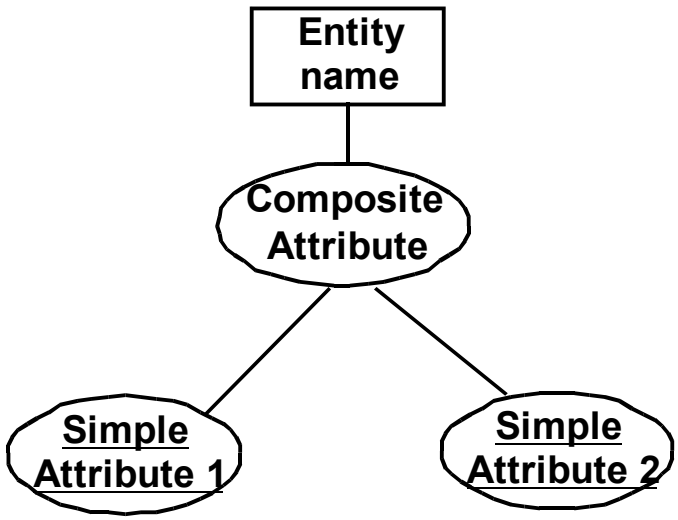
Here is a generic entity structure using ER notation:



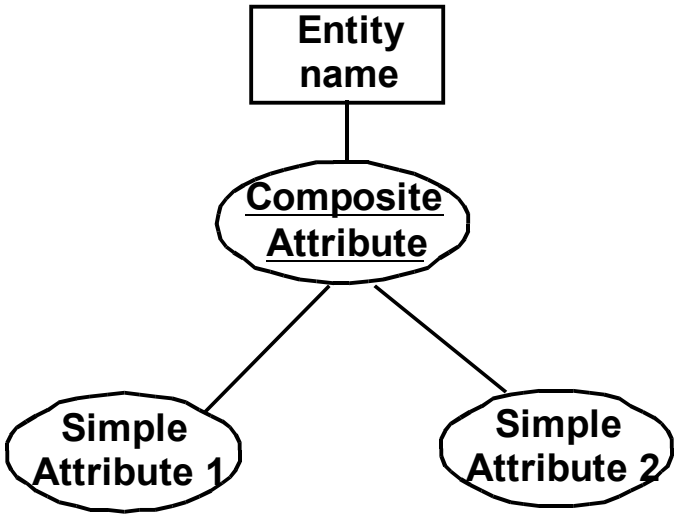
Notes:

- Rectangles are used to represent *entity types.* These correspond to *object types* in object-oriented modelling.
- *Attributes* are represented by ovals.
- *Composite attributes* are those which have sub-attributes.
- *Simple attributes* have no sub-attributes.
- *Key attributes* are underlined. A key attribute is one whose value corresponds to only one instance of the entity type.

- Note that *every* underlined attribute is a key. Thus, in the diagram below, both *Simple Attribute 1* and *Simple Attribute 2* are keys, individually.

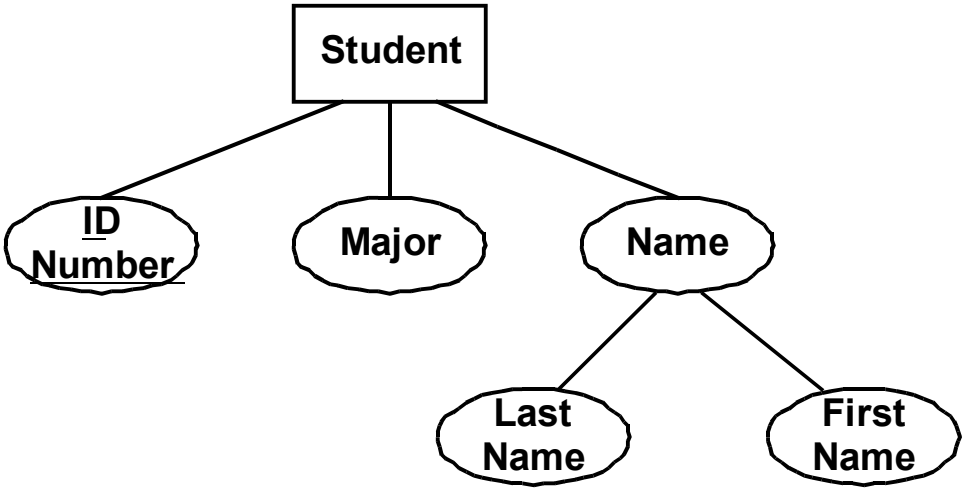


- To obtain a representation in which the two attributes together form a key, a composite attribute must be used as the key, as illustrated below.



It is important to distinguish *types* from *instances.*

Example: Here is the student type again:



Here is an instance of this class:

```
Student:
    ID_number: 751030-0123;
    Major: Computing Science;
    Name:
        Last_name:  Nordmann;
        First name: Kari;
End Student.
```

Note: The syntax of this instance definition is not part of ER modelling, and is inconsequential.

Bells and Whistles:

Null values:

- An attribute may fail to have a value for at least two reasons:
  - The value is missing. This is called a *missing value* null.
  - The value does not exist. This is called a *not applicable* null.

  - A popular example which covers both cases is that of the telephone number of a student.
    - A missing-value null indicates that the telephone number is unknown.
    - A not-applicable null indicates that the student does not have a telephone.

Note that a key attribute may never be null.

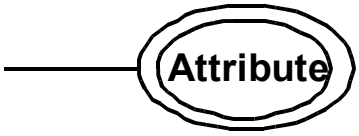
Multiple values:

- An attribute may have the possibility of containing many values.

  A popular example is that of the children of an employee. The proper way to view such an attribute is that its value is a *set* individuals.

  An example focused on the student database is that of the courses taken by a given student.

  Multi-valued attributes are depicted using a double oval:



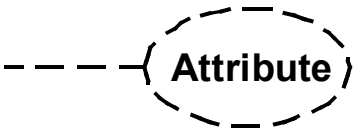
Derived attributes:

Sometimes, the value of an attribute may be computed from the values of other attributes. Such attributes are termed *derived.*

A popular example is that of age, which may be computed from the attribute birthdate.
This example is slightly contrived, because the current date must also be part of the database.

In Sweden, a better example is that of determining gender from personnummer. In a Swedish personnummer, the second-to-last digit is always even for a female, and odd for a male.

The graphical representation is via dashed figures.

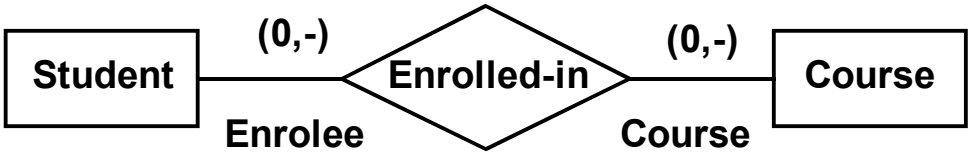


Relationships:

As suggested by the name, Entity-Relationship modelling involves not only entities, but relationships between them as well.

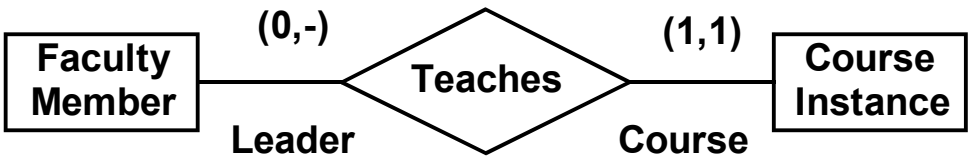
A relationship is represented by a diamond.

For example,



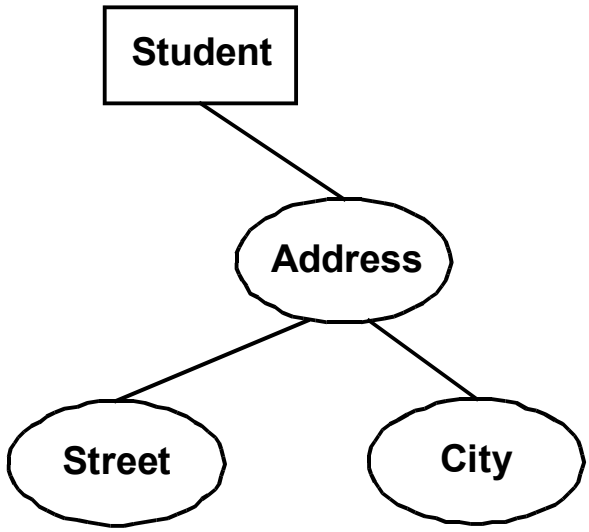
The numbers in parentheses indicate the minimum and maximum number of entity objects which may participate in the relationship.

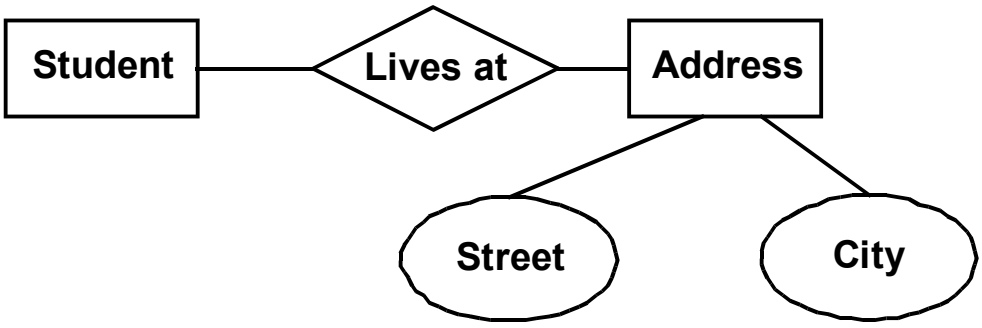
- (m,n) means: minimum=m; maximum=n.
- (m,–) means: minimum=m; no maximum.



Relationships may also be of higher degree (*e.g.* ternary), but this is less common, and results in much more complex relational models.

Note that there is some flexibility as to whether something is modelled with compound attributes or with a relationship.





Weak entity types:

- A *weak entity type* is one which does not have its own key attributes.
- These are usually represented by double rectangles.
- For an example, look at the Dependent entity of the Company database of the textbook (Figure 3.15).

Remark on notation:

- In the textbook, two alternative notations for ER diagrams, as represented in Figure 3.2 and Figure 3.15.

- In this course, the notation of Figure 3.15 will be used. (It seems far more logical and systematic.)