The ER data model employs 3 basic concepts:

* + entity sets,
    - entity: object that is distinguishable from other objects
    - entity set: set of entities of same type that share same properties (table)
    - an entity is represented by set of attributes
    - A subset of the attributes form a **primary key** of the entity set

A close-up of a white board

Description automatically generated

* + relationship sets,
    - relationship: association among several entities



* + - relationship set: mathematical relation among n>=2 entities, each taken from entity sets
      * {(*e*1, *e*2, … *en*) | *e*1 ∈ *E*1, *e*2 ∈ *E*2, …, *en* ∈ *En*}
      * (e1, e2, …, en) is a relationship
      * Example:

(44553,22222) ∈ *advisor*

* + - * Example: we define the relationship set *advisor* to denote the associations between students and the instructors who act as their advisors.
      * Pictorially, we draw a line between related entities.
      * 
      * Diamonds represent relationship sets.
      * 
      * An attribute can also be associated with a relationship set.
      * For instance, the *advisor* relationship set between entity sets *instructor* and *student* may have the attribute *date* which tracks when the student started being associated with the advisor
      * 
      * 
      * Entity sets of a relationship need not be distinct
        + Each occurrence of an entity set plays a “role” in the relationship
      * The labels “*course\_id*” and “*prereq\_id*” are called **roles**.
      * A diagram of a course

        Description automatically generated
      * Binary relationship
        + involve 2 entity sets
      * most relationships are binary. more than 2 entity sets rare:
        + example: students work on research projects under the guidance of an instructor
        + relationship project\_guide is a ternary relationship between instructor-student-project (ternary)
        + 

* + attributes.
    - simple – composite attributes



* + - single valued – multivalued attributes {}
    - derived attributes(): can be computed from other attributes - stored
    - domain: set of permitted values for each attribute
    - Complex attributes in ER diagram (multivalued & composite)



* Ignoring multivalued attributes, extended instructor schema is
  + *instructor(ID,   
     first\_name, middle\_initial, last\_name,  
     street\_number, street\_name,   
     apt\_number, city, state, zip\_code,   
     date\_of\_birth)*

**Mapping cardinality constraints**

Express the number of entities to which another entity can be associated via a relationship set.

For a binary relationship set the mapping cardinality must be one of the following types:

* + One to one
  + One to many
  + Many to one
  + Many to many

A diagram of a number of objects

Description automatically generated with medium confidence

A diagram of a number of objects

Description automatically generated with medium confidence

We express cardinality constraints by drawing either a directed line (→), signifying “one,” or an undirected line (—), signifying “many,” between the relationship set and the entity set.

ONE-TO-ONE

A student is associated with at most one *instructor* via the relationship *advisor*

A *student* is associated with at most one *instructor*  via *advisor*



ONE-TO-MANY

an instructor is associated with several (including 0) students via *advisor*

a student is associated with at most one instructor via advisor,



MANY-TO-ONE

an instructor is associated with at most one student via *advisor*,

and a student is associated with several (including 0) instructors via *advisor*



MANY-TO-MANY



An instructor is associated with several (possibly 0) students via *advisor*

A student is associated with several (possibly 0) instructors via *advisor*



total participation:

2 çizgi ile gösterilir

every entity in the entity set participates in at least one relationship in the relationship set



participation of *student* in *advisor r*elation is total

every *student* must have an associated instructor

Partial participation:

some entities may not participate in any relationship in the relationship set

* + Example: participation of *instructor* in *advisor* is partial

L … H cardinality

L=1 🡪 total participation

H=1 🡪 entity participates in at most one relationship

H=\* 🡪 no limit



Instructor can advise 0 or more students. A student must have 1 advisor; cannot have multiple advisors

**Cardinality constraints on ternary relationship**



We allow at most one arrow out of a ternary (or greater degree) relationship to indicate a cardinality constraint

For example, an arrow from *proj\_guide* to *instructor* indicates each student has at most one guide for a project

A primary key for an entity is a set of attributes that suffice to distinguish entities from each other

The primary key for R relationship set is consists of the union of the primary keys of entity sets E1, E2, ..En

* Example: relationship set “advisor”.
  + The primary key consists of *instructor.ID* and s*tudent.ID*

Choice of primary key:

* Many-to-Many relationships. The preceding union of the primary keys is a minimal superkey and is chosen as the primary key.
* One-to-Many relationships . The primary key of the “Many” side is a minimal superkey and is used as the primary key.
* Many-to-one relationships. The primary key of the “Many” side is a minimal superkey and is used as the primary key.
* One-to-one relationships. The primary key of either one of the participating entity sets forms a minimal superkey, and either one can be chosen as the primary key.

**WEAK ENTITY SETS**

section – course -------------> related entities

sec\_course ---> relationship set between entity sets section and course

section’da course id var, sec\_course’daki bilgi gereksiz

sec\_course’tan kurtulabiliriz, ama bu durumda ilişki implicit in an attribute olur, bunu istemeyiz

o zaman course\_id’yi section’dan kaldıralım?

* O zaman da section bir row’u unique olarak tanımlayamaz, yeterli attribute yok

sec\_course’a ekstra bilgi veren özel bir relationship olarak davranırız. Bu durumda course\_id section entitylerini unique olarak tanımlamalı

A **weak entity set** is one whose existence is dependent on another entity, called its **identifying entity**

Instead of associating a primary key with a weak entity, we use the identifying entity, along with extra attributes called **discriminator** to uniquely identify a weak entity.

Every weak entity must be associated with an identifying entity; that is, the weak entity set is said to be **existence dependent** on the identifying entity set.

The identifying entity set is said to **own** the weak entity set that it identifies.

The relationship associating the weak entity set with the identifying entity set is called the **identifying relationship**.

ER:

* weak entity set double rectangle ile gösterilir
* discriminator of weak entity set is underlined with dashed line
* relationship set connecting the weak entity set to the identifying strong entity set is depicted by a double diamond.
* Primary key for *section* – (*course\_id, sec\_id, semester, year*)
* 





A diagram of a computer

Description automatically generated

Example: Multivalued attribute *phone\_number* of *instructor* is represented by a schema:  
 *inst\_phone=* ( *ID, phone\_number*)

A diagram of a diagram

Description automatically generated

Specialization

Bir varlık türünün, daha özel veya daha spesifik alt varlık türlerine bölünmesini ifade eder.

* **Attribute inheritance** – a lower-level entity set inherits all the attributes and relationship participation of the higher-level entity set to which it is linked.

A diagram of a person

Description automatically generated

Representing specialization via schemas

* Form a schema for the higher-level entity
* Form a schema for each lower-level entity set, include primary key of higher-level entity set and local attributes
* A close-up of a list of attributes

  Description automatically generated
* Drawback: getting information about, an *employee* requires accessing two relations, the one corresponding to the low-level schema and the one corresponding to the high-level schema
* Form a schema for each entity set with all local and inherited attributes
* A close-up of text

  Description automatically generated
* Drawback: *name, street* and *city* may be stored redundantly for people who are both students and employees

GENERALIZATION

* **A bottom-up design process** – combine a number of entity sets that share the same features into a higher-level entity set.
* The terms specialization and generalization are used interchangeably.

**Completeness constraint**

* **Completeness constraint** -- specifies whether or not an entity in the higher-level entity set must belong to at least one of the lower-level entity sets within a generalization.

**Total Completeness Constraint**

Örneğin, "müşteri" ve "sipariş" arasındaki ilişki, her iki varlık türü için de zorunlu olabilir, çünkü her siparişin bir müşteriye ait olması ve her müşterinin en az bir siparişi olması gerekir.

**Partial Completeness Constraint**

Örneğin, "müşteri" ve "sipariş" arasındaki ilişki, müşterilerin sipariş vermek zorunda olmadığı durumu ifade edebilir.



A diagram of a diagram

Description automatically generated

convert non-binary to binary relationship



