4/1

There will be a quiz in this week’s initial discussion sessions

You must attend the session you registered for

Database is a very large, integrated collection of data

Usually a model of a real world enterprise

Entities(student, courses, facebook…)

Relationship( Susan is taking CS234, susan is a friend of Lynn,..)

Database management system(DBMS)

DBMS

Data independenve

Efficient data access

Reduced application development time

Data integrity and security

4/3

1. Internal Level: Actual PHYSICAL storage structure and access paths.
2. Conceptual or Logical Level: Structure and constraints for the entire database
3. External or View level: Describes various user views



Data Independence

Application are isulated from how data is actually structured and stored, thanks to schema layering and high-level queries

Logical data independence: Protection from change in the logical structure of data

Physical data independence:

Data independence: <https://en.wikipedia.org/wiki/Data_independence>

4/5

The first assignment available

Deparment must have at least one manager, or can not exist

Underline means unique

The ER model defines the conceptual view of a database. It works around real-world entities and the associations among them. At view level, the ER model is considered a good option for designing databases.

## Entity

An entity can be a real-world object, either animate or inanimate, that can be easily identifiable. For example, in a school database, students, teachers, classes, and courses offered can be considered as entities. All these entities have some attributes or properties that give them their identity.

An entity set is a collection of similar types of entities. An entity set may contain entities with attribute sharing similar values. For example, a Students set may contain all the students of a school; likewise a Teachers set may contain all the teachers of a school from all faculties. Entity sets need not be disjoint.

## Attributes

Entities are represented by means of their properties, called **attributes**. All attributes have values. For example, a student entity may have name, class, and age as attributes.

There exists a domain or range of values that can be assigned to attributes. For example, a student's name cannot be a numeric value. It has to be alphabetic. A student's age cannot be negative, etc.

### Types of Attributes

* **Simple attribute** − Simple attributes are atomic values, which cannot be divided further. For example, a student's phone number is an atomic value of 10 digits.
* **Composite attribute** − Composite attributes are made of more than one simple attribute. For example, a student's complete name may have first\_name and last\_name.
* **Derived attribute** − Derived attributes are the attributes that do not exist in the physical database, but their values are derived from other attributes present in the database. For example, average\_salary in a department should not be saved directly in the database, instead it can be derived. For another example, age can be derived from data\_of\_birth. 可以被计算出来的
* **Single-value attribute** − Single-value attributes contain single value. For example − Social\_Security\_Number.
* **Multi-value attribute** − Multi-value attributes may contain more than one values. For example, a person can have more than one phone number, email\_address, etc.

These attribute types can come together in a way like −

* simple single-valued attributes
* simple multi-valued attributes
* composite single-valued attributes
* composite multi-valued attributes

### Entity-Set and Keys

Key is an attribute or collection of attributes that uniquely identifies an entity among entity set.

For example, the roll\_number of a student makes him/her identifiable among students.

* **Super Key** − A set of attributes (one or more) that collectively identifies an entity in an entity set.
* **Candidate Key** − A minimal super key is called a candidate key. An entity set may have more than one candidate key.
* **Primary Key** − A primary key is one of the candidate keys chosen by the database designer to uniquely identify the entity set.

## Relationship

The association among entities is called a relationship. For example, an employee **works\_at** a department, a student **enrolls** in a course. Here, Works\_at and Enrolls are called relationships.

### Relationship Set

A set of relationships of similar type is called a relationship set. Like entities, a relationship too can have attributes. These attributes are called **descriptive attributes**.

### Degree of Relationship

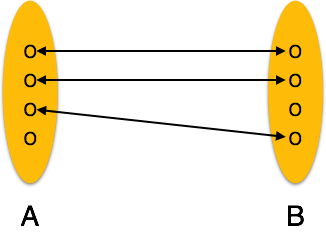
The number of participating entities in a relationship defines the degree of the relationship.

* Binary = degree 2
* Ternary = degree 3
* n-ary = degree

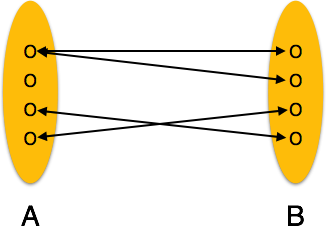
### Mapping Cardinalities

**Cardinality** defines the number of entities in one entity set, which can be associated with the number of entities of other set via relationship set.

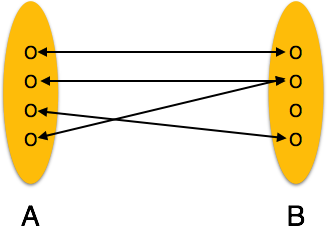
* **One-to-one** − One entity from entity set A can be associated with at most one entity of entity set B and vice versa.



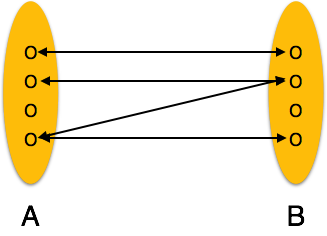
* **One-to-many** − One entity from entity set A can be associated with more than one entities of entity set B however an entity from entity set B, can be associated with at most one entity.



* **Many-to-one** − More than one entities from entity set A can be associated with at most one entity of entity set B, however an entity from entity set B can be associated with more than one entity from entity set A.



* **Many-to-many** − One entity from A can be associated with more than one entity from B and vice versa.



Entity Relationship Diagram

Let us now learn how the ER Model is represented by means of an ER diagram. Any object, for example, entities, attributes of an entity, relationship sets, and attributes of relationship sets, can be represented with the help of an ER diagram.

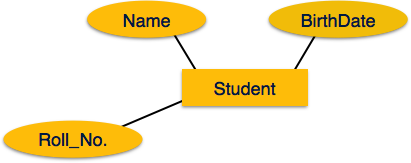
Entity

Entities are represented by means of rectangles. Rectangles are named with the entity set they represent.

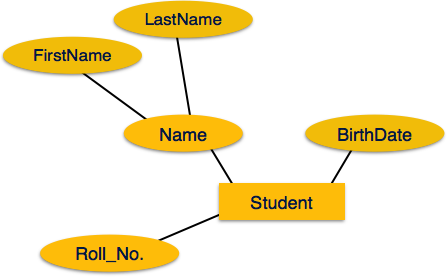
Entities in a school database

Attributes

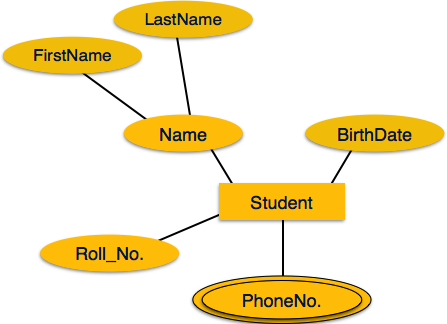
Attributes are the properties of entities. Attributes are represented by means of ellipses. Every ellipse represents one attribute and is directly connected to its entity (rectangle).



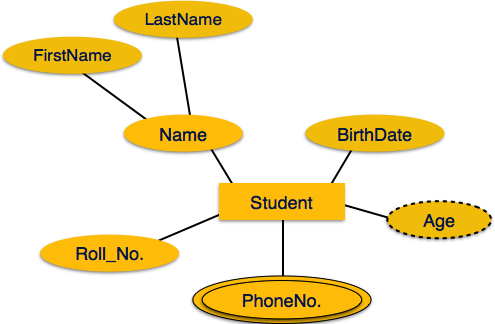
If the attributes are **composite**, they are further divided in a tree like structure. Every node is then connected to its attribute. That is, composite attributes are represented by ellipses that are connected with an ellipse.



**Multivalued** attributes are depicted by double ellipse. 可以多个的话是双层椭圆



**Derived** attributes are depicted by dashed ellipse. 虚线椭圆代表是可以算出来的



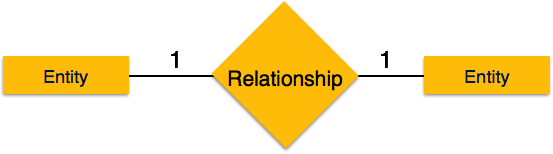
Relationship

Relationships are represented by diamond-shaped box. Name of the relationship is written inside the diamond-box. All the entities (rectangles) participating in a relationship, are connected to it by a line.

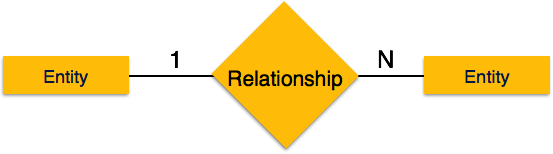
Binary Relationship and Cardinality

A relationship where two entities are participating is called a **binary relationship**. Cardinality is the number of instance of an entity from a relation that can be associated with the relation.

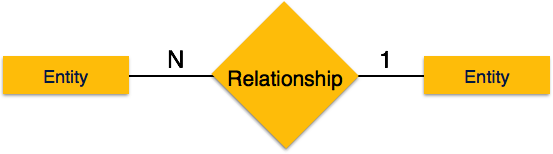
* **One-to-one** − When only one instance of an entity is associated with the relationship, it is marked as '1:1'. The following image reflects that only one instance of each entity should be associated with the relationship. It depicts one-to-one relationship.



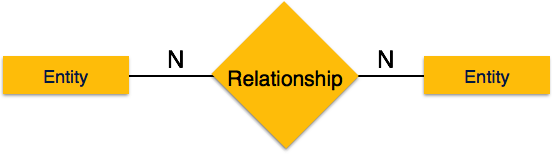
* **One-to-many** − When more than one instance of an entity is associated with a relationship, it is marked as '1:N'. The following image reflects that only one instance of entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts one-to-many relationship.



* **Many-to-one** − When more than one instance of entity is associated with the relationship, it is marked as 'N:1'. The following image reflects that more than one instance of an entity on the left and only one instance of an entity on the right can be associated with the relationship. It depicts many-to-one relationship.

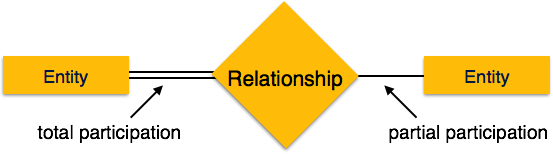


* **Many-to-many** − The following image reflects that more than one instance of an entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts many-to-many relationship.



Participation Constraints

* **Total Participation** − Each entity is involved in the relationship. Total participation is represented by double lines.
* **Partial participation** − Not all entities are involved in the relationship. Partial participation is represented by single lines.



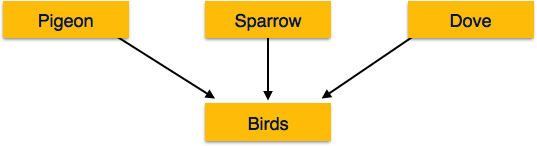
In a [relational database](https://en.wikipedia.org/wiki/Relational_database), a **weak entity** is an entity that cannot be uniquely identified by its attributes alone; therefore, it must use a [foreign key](https://en.wikipedia.org/wiki/Foreign_key) in conjunction with its attributes to create a [primary key](https://en.wikipedia.org/wiki/Primary_key). The foreign key is typically a primary key of an entity it is related to.

The ER Model has the power of expressing database entities in a conceptual hierarchical manner. As the hierarchy goes up, it generalizes the view of entities, and as we go deep in the hierarchy, it gives us the detail of every entity included.

Going up in this structure is called **generalization**, where entities are clubbed together to represent a more generalized view. For example, a particular student named Mira can be generalized along with all the students. The entity shall be a student, and further, the student is a person. The reverse is called **specialization** where a person is a student, and that student is Mira.

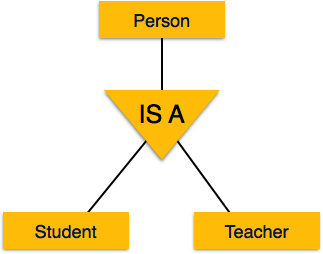
Generalization

As mentioned above, the process of generalizing entities, where the generalized entities contain the properties of all the generalized entities, is called generalization. In generalization, a number of entities are brought together into one generalized entity based on their similar characteristics. For example, pigeon, house sparrow, crow and dove can all be generalized as Birds.



Specialization

Specialization is the opposite of generalization. In specialization, a group of entities is divided into sub-groups based on their characteristics. Take a group ‘Person’ for example. A person has name, date of birth, gender, etc. These properties are common in all persons, human beings. But in a company, persons can be identified as employee, employer, customer, or vendor, based on what role they play in the company.

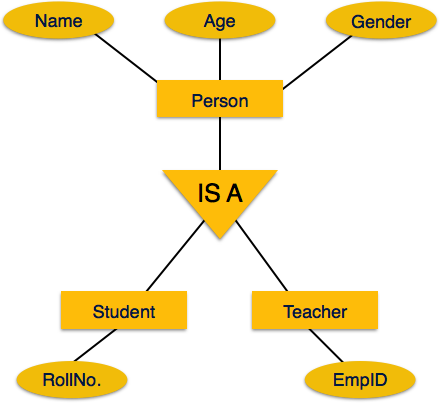


Similarly, in a school database, persons can be specialized as teacher, student, or a staff, based on what role they play in school as entities.

Inheritance

We use all the above features of ER-Model in order to create classes of objects in object-oriented programming. The details of entities are generally hidden from the user; this process known as **abstraction**.

Inheritance is an important feature of Generalization and Specialization. It allows lower-level entities to inherit the attributes of higher-level entities.



For example, the attributes of a Person class such as name, age, and gender can be inherited by lower-level entities such as Student or Teacher.

下滑虚线装weak entity

Ternary Relationship (三角恋)

General note: Relationship Key <= (entity key)

ISA (is a) Hierarchies

Aggregation allows us to treat the relationship set as a whole

Lec 4/8

All of entity should has its identify key

No need to worry about how fast the query code you write, the system going to change it to its fastest way to find the result

Select star 全部

Lec 4/10

Key(has to be unique):

Primary key

Candidate key

Superkey {xx, yy}, student id is a key, and with others attribute will become a super key

Foreign key

Primary key need to be decided before the database finished

Overlap constraint: 可以是其中一种，也可以是多种/ 只可以是一种

Covering constraint： 现有种类是父族的所有/不是所有

LEC 4/12/19

A view is just a relation, but we store its definition rather than storing the (materialized) set of tuples

只是给别人看看这个relationship，不存或者做任何实际动作

N side needs 1 side

Lec 4/15

Good design due to elimination of redundancy

一个table里每个相同的attribute与到另外一个attrribute也是每个相同的，那可以建一个新的小table

Lec 4/17/19

Midterm is a week from next Monday

2 sides cheat ship

If no FDs hold, then there is no redundancy,如果每个种类的数据跟其他种类的数据都没有固定relationship，那么它里面是不存在redundancy的

1NF 其attribute只有一个

Rel’n R is in 2NF if it is in 1NF and no non-prime attribute is partially dependent on a candidate key of R

On delete cascade:

ISA 需要

Weak entity 需要

Total participation 需要delete

More than one attribute 需要 delete

Not Null:

ENUM

有primary key就不需要null

Null:

Optional

1 对多或者多对一或者一对一的菱形可以不要写table

多的那边要把一的key放进去，一的那边不用

Table 里面写了forgreign key，别人要用你的forreign key的时候就是相当于在用你的key

Fri Lec 4/19/19

Candidate Key – A Candidate Key can be any column or a combination of columns that can qualify as unique key in database. There can be multiple Candidate Keys in one table. Each Candidate Key can qualify as Primary Key.

Primary Key – A Primary Key is a column or a combination of columns that uniquely identify a record. Only one Candidate Key can be Primary Key.



If you have more than one candidate key, one of them has to be primary key, others will be unique key

Midterm: Cheat shift

Notcover ch11

Lec 5/1