Database Design

Topics to be Covered

- Database design & E-R Model
- Entity–Relationship model (E-R model)
- E-R Diagrams-Constraints
- Extended E-R features

Database Design

- The database designer needs to interact extensively with domain experts and specify the user requirements
- The designer chooses a data model and, by applying the concepts of the chosen data model, translates these requirements into a conceptual schema of the database
- At the stage of conceptual design, the designer can review the schema to ensure it meets functional requirements

Database Design (Contd..)

- Logical-design phase
  - The conceptual schema defined using the entity-relationship model is converted into a relation schema in this phase
- Physical-design phase
  - The physical features of the database are specified in this phase

ER Model

- The Entity Relational Model is a model for identifying entities to be represented in the database and representation of how those entities are related
- The ER data model specifies enterprise schema that represents the overall logical structure of a database graphically

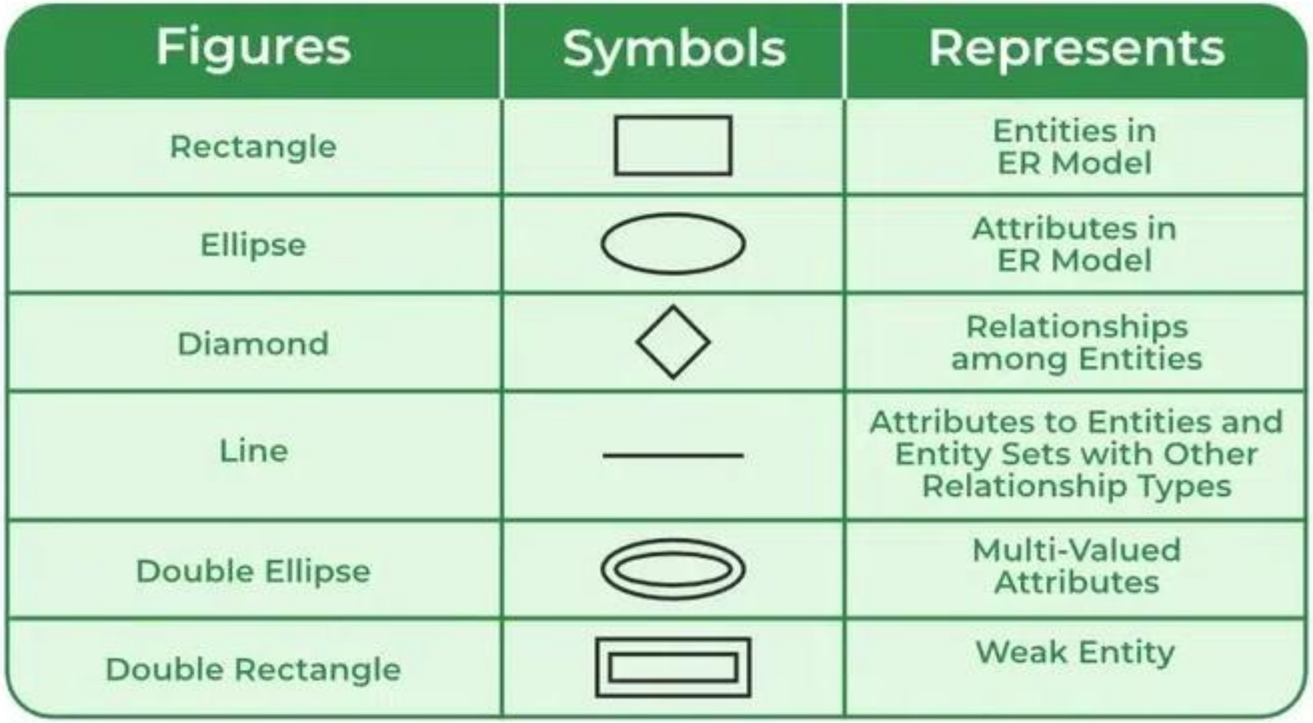
Why Use ER Diagrams In DBMS?

- ER diagrams are used to represent the E-R model in a database, which makes them easy to convert into relations (tables)
- ER diagrams provide the purpose of real-world modeling of objects which makes them intently useful
- ER diagrams require no technical knowledge and no hardware support
- These diagrams are very easy to understand and easy to create even for a naive user
- It gives a standard solution for visualizing the data logically

Symbols Used in ER Model

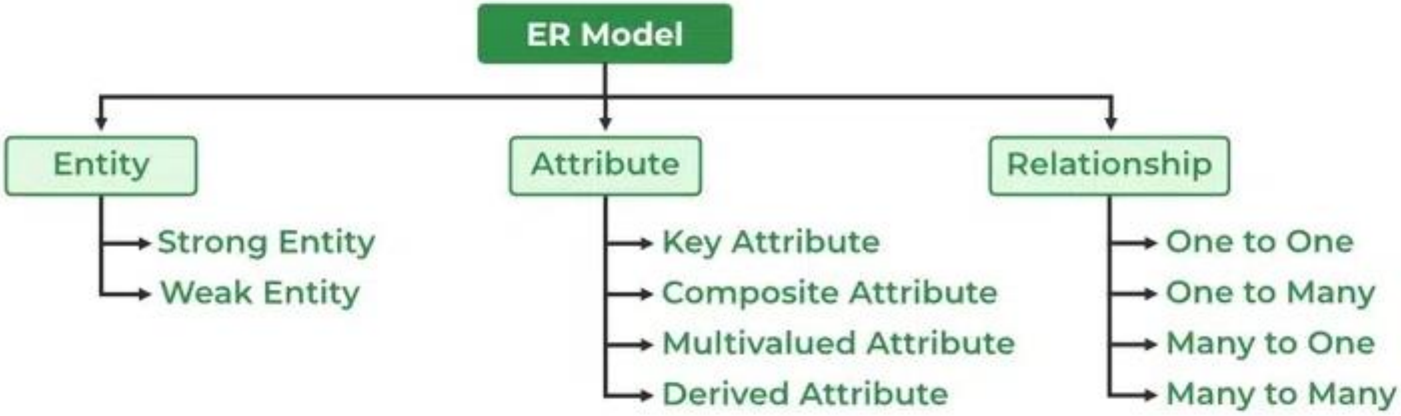
- **Rectangles:** Rectangles represent Entities in the ER Model
- **Ellipses:** Ellipses represent Attributes in the ER Model
- **Diamond:** Diamonds represent Relationships among Entities
- **Lines:** Lines represent attributes to entities and entity sets with other relationship types
- **Double Ellipse:** Double Ellipses represent Multi-Valued Attributes
- **Double Rectangle:** Double Rectangle represents a Weak Entity

Symbols used in ER model



Figures	Symbols	Represents
Rectangle	▭	Entities in ER Model
Ellipse	⬭	Attributes in ER Model
Diamond	◇	Relationships among Entities
Line	—	Attributes to Entities and Entity Sets with Other Relationship Types
Double Ellipse	◎	Multi-Valued Attributes
Double Rectangle	⧈	Weak Entity

Components of ER Model


ER Model

Entity

- Strong Entity
- Weak Entity

Attribute

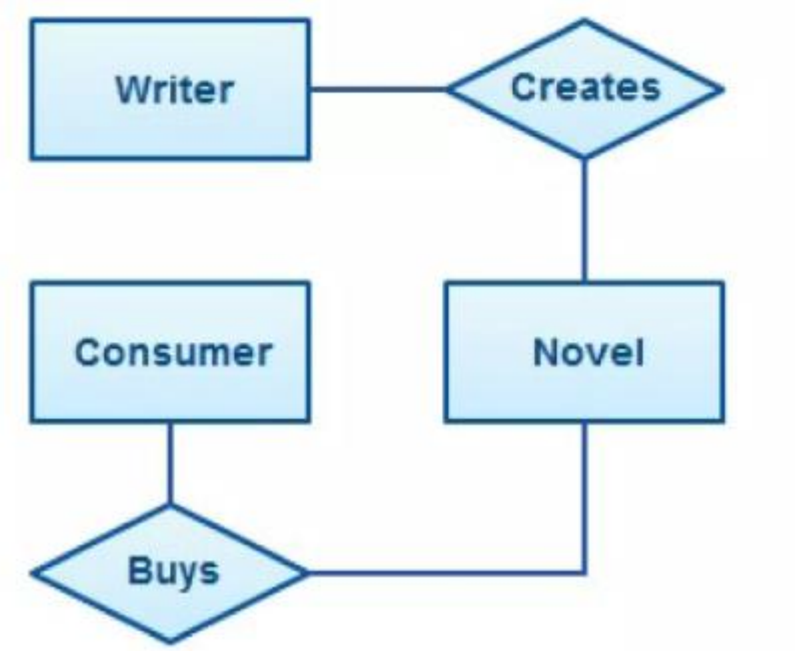
- Key Attribute
- Composite Attribute
- Multivalued Attribute
- Derived Attribute

Relationship

- One to One
- One to Many
- Many to One
- Many to Many

Example

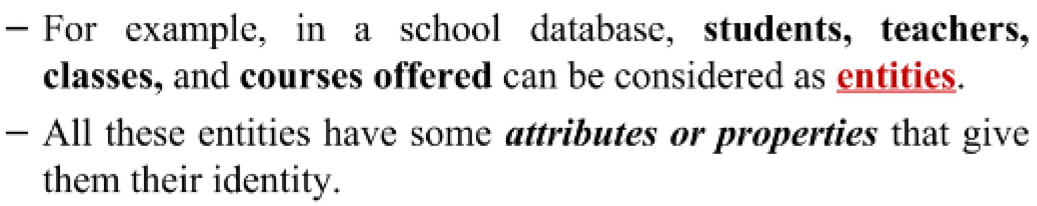
For example, the elements writer, novel, and consumer may be described using ER diagrams this way:



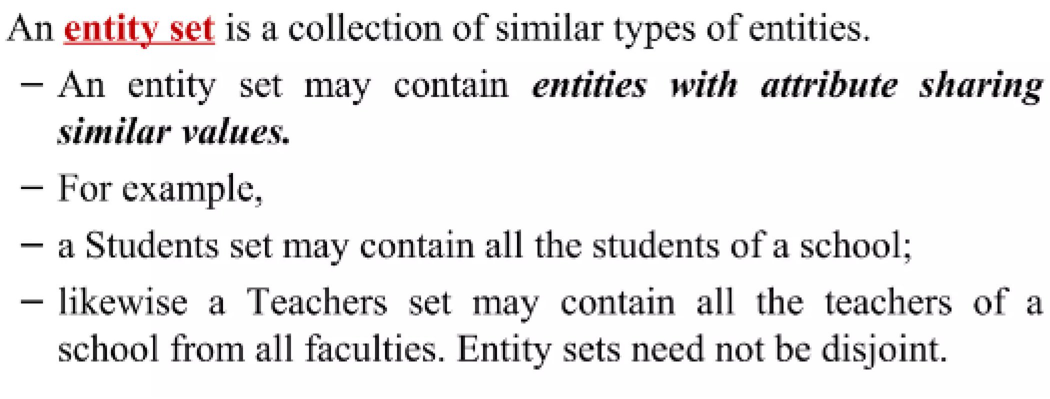
The elements inside **rectangles** are called **<u>entities</u>** while the items inside **diamonds** denote the **<u>relationships</u>** between entities.

Entity

Entity can be any real world object



- 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.

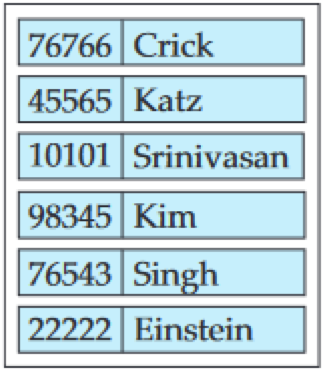


An **entity set** is a collection of similar types of entities.

- 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.

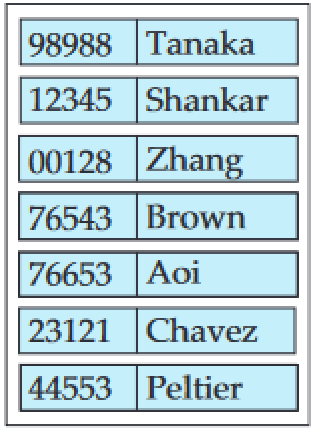
Entity Set

An entity set is a set of entities of the same type that share the same properties, or attributes.



76766	Crick
45565	Katz
10101	Srinivasan
98345	Kim
76543	Singh
22222	Einstein

instructor



98988	Tanaka
12345	Shankar
00128	Zhang
76543	Brown
76653	Aoi
23121	Chavez
44553	Peltier

student

Strong Entity

- A strong entity is not dependent on any other entity in the schema
- A strong entity will always have a primary key
- Strong entities are represented by a single rectangle
- The relationship of two strong entities is represented by a single diamond
- Various strong entities, when combined together, create a strong entity set

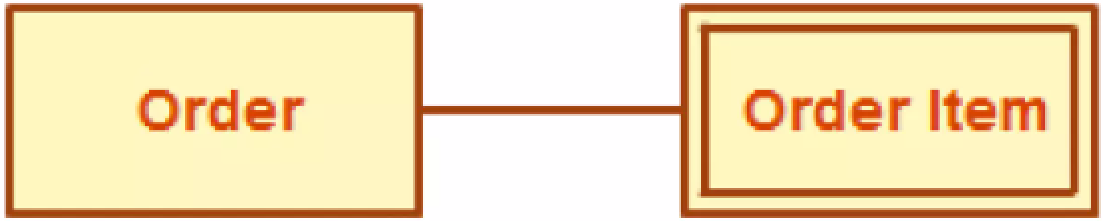
Weak entity

- A weak entity is dependent on a strong entity to ensure its existence
- Unlike a strong entity, a weak entity does not have any primary key
- It instead has a partial discriminator key
- A weak entity is represented by a double rectangle
- The relation between one strong and one weak entity is represented by a double diamond

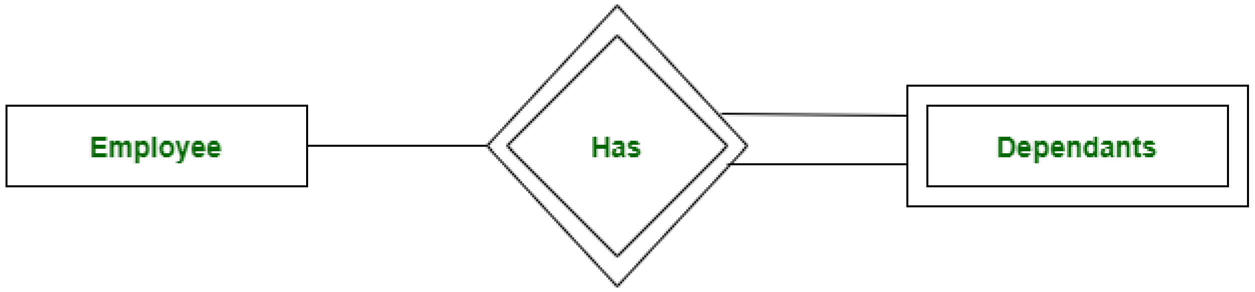
Example-1

An entity like **order item** is a good example for this.

- The order item will be meaningless without an order so it depends on the existence of order.

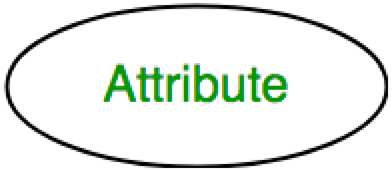


Example-2



Attributes

Attributes are the properties that define the entity type.
For example, Roll_No, Name, DOB, Age, Address, and Mobile_No are the attributes that define entity type Student.

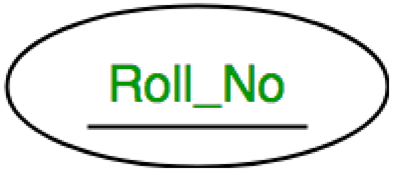


Key Attribute

The attribute which **uniquely identifies each entity** in the entity set is called the key attribute.

For example, Roll_No will be unique for each student

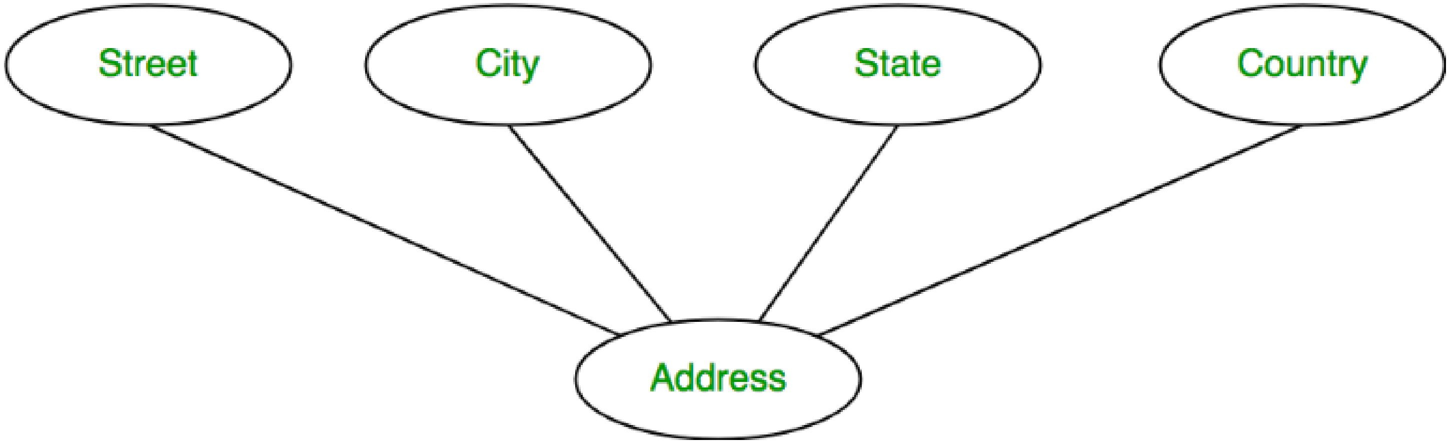
In ER diagram, the key attribute is represented by an oval with underlying lines



Composite Attribute

- An attribute composed of many other attributes is called a composite attribute
- For example, the Address attribute of the student Entity type consists of Street, City, State, and Country
- In ER diagram, the composite attribute is represented by an oval comprising of ovals

Composite Attribute (Contd..)

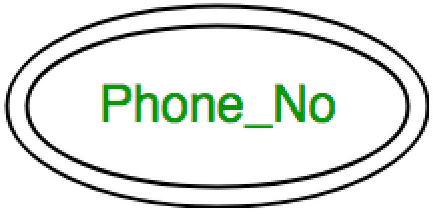


Multivalued Attribute

An attribute consisting of more than one value for a given entity

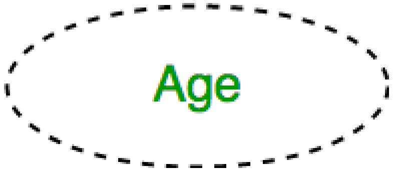
For example, Phone_No (can be more than one for a given student)

In ER diagram, a multivalued attribute is represented by a double oval

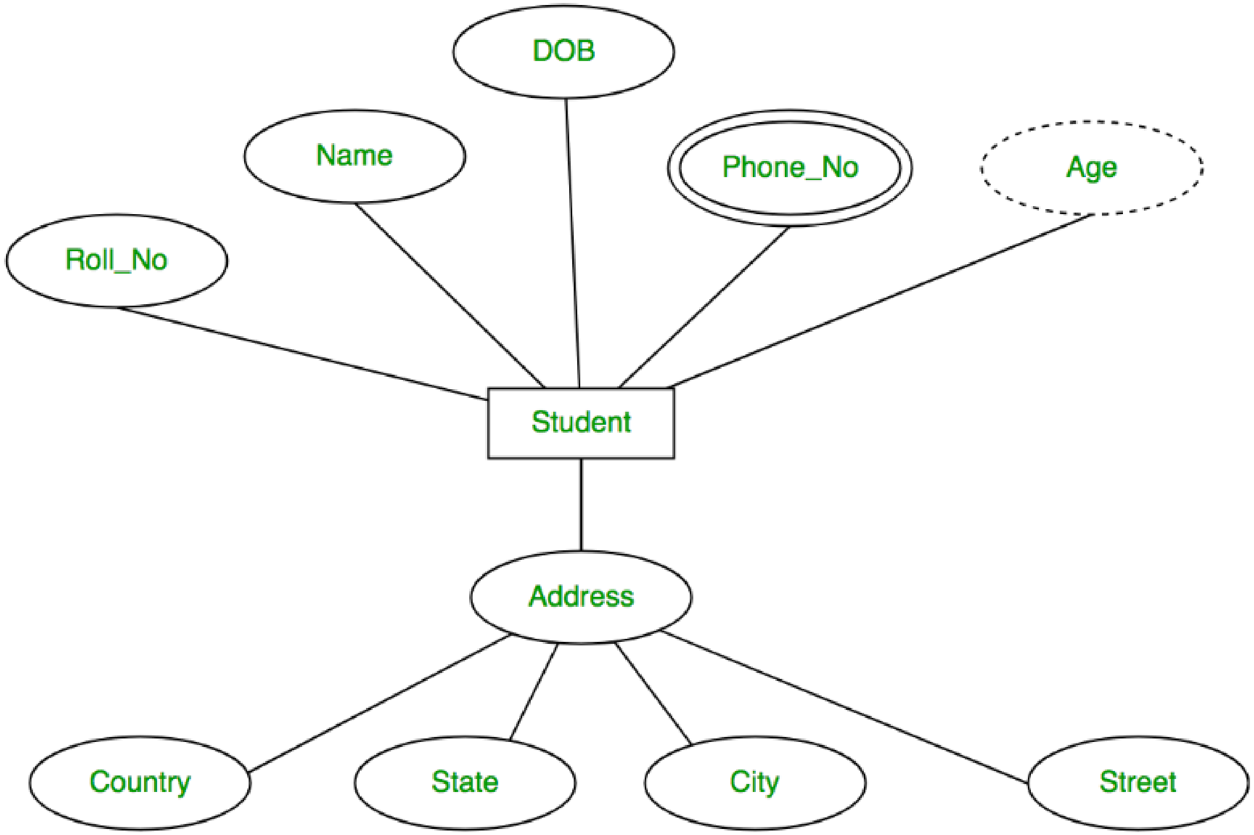


Derived Attribute

- An attribute that can be derived from other attributes of the entity type is known as a derived attribute
- E.g.; Age (can be derived from DOB)
- In ER diagram, the derived attribute is represented by a dashed oval



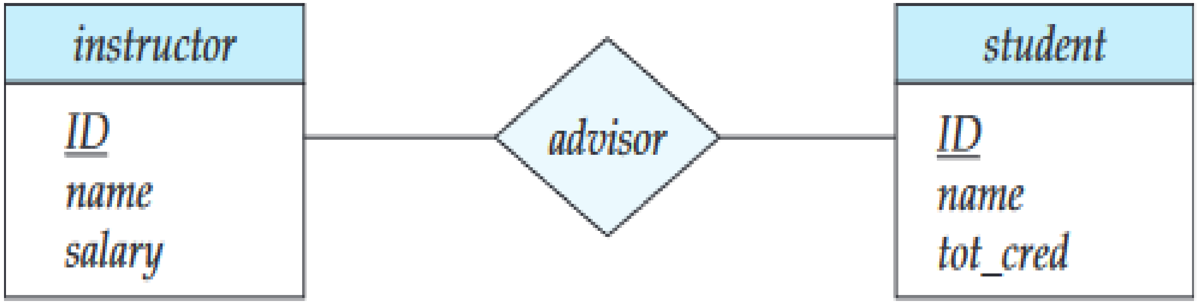
Example
Student Entity Set with its attributes



Entity and Attributes

Entity-Relationship Diagram

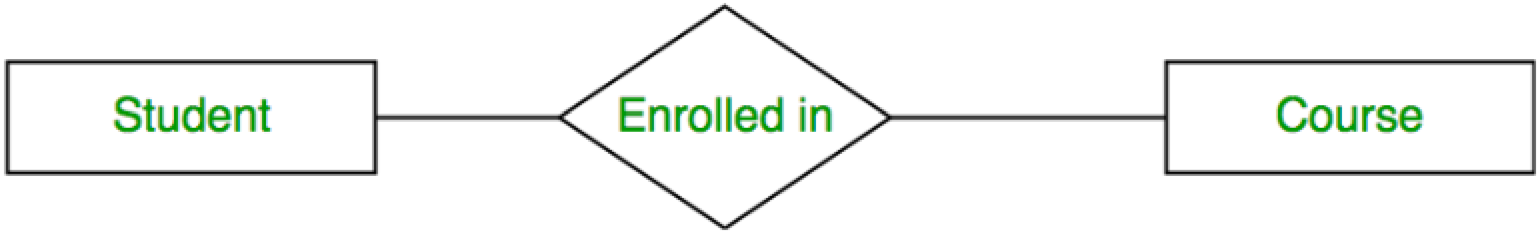
ER diagram corresponding to instructors and students



Relationship Type

- A Relationship Type represents the association between entity types
- For example, ‘Enrolled in’ is a relationship type that exists between entity type Student and Course
- In ER diagram, the relationship type is represented by a diamond and connecting the entities with lines

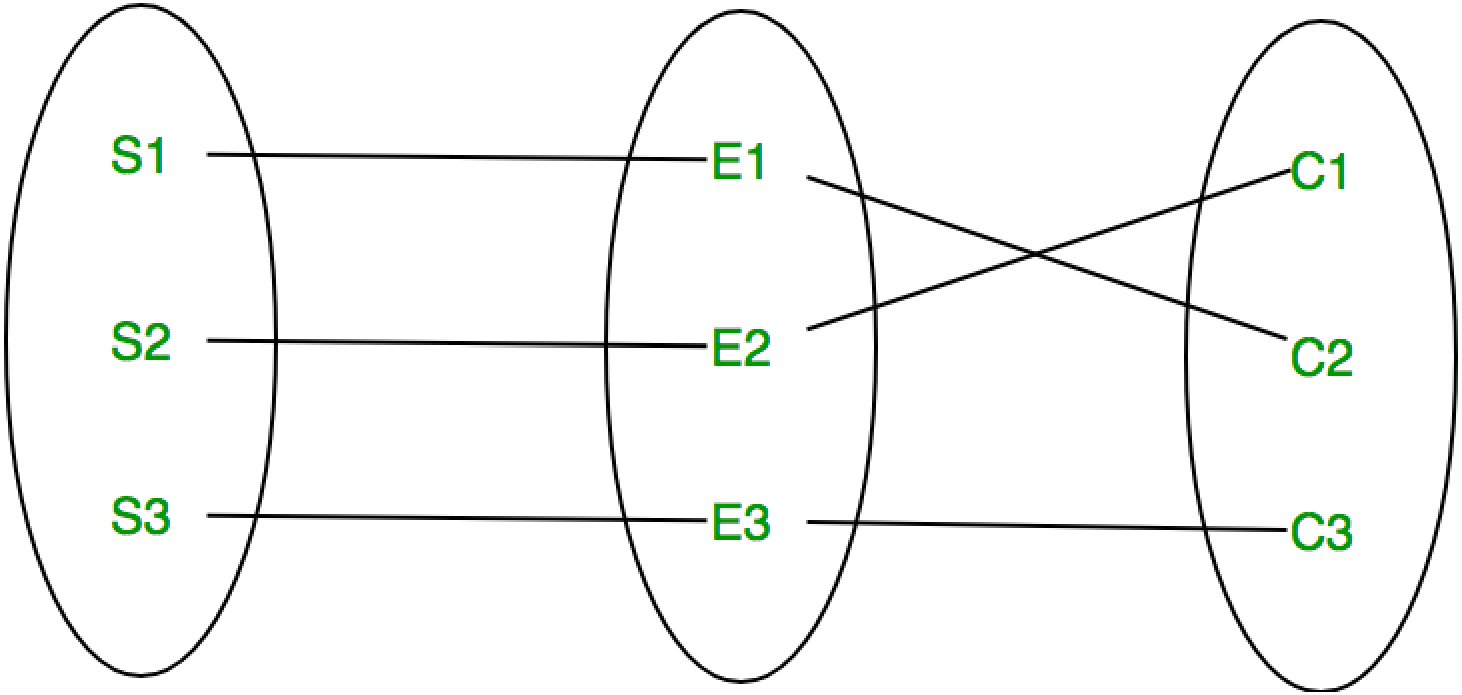
Example



Relationship Set

- A set of relationships of the same type is known as a relationship set
- The following relationship set depicts S1 as enrolled in C2, S2 as enrolled in C1, and S3 as registered in C3

Relationship Set

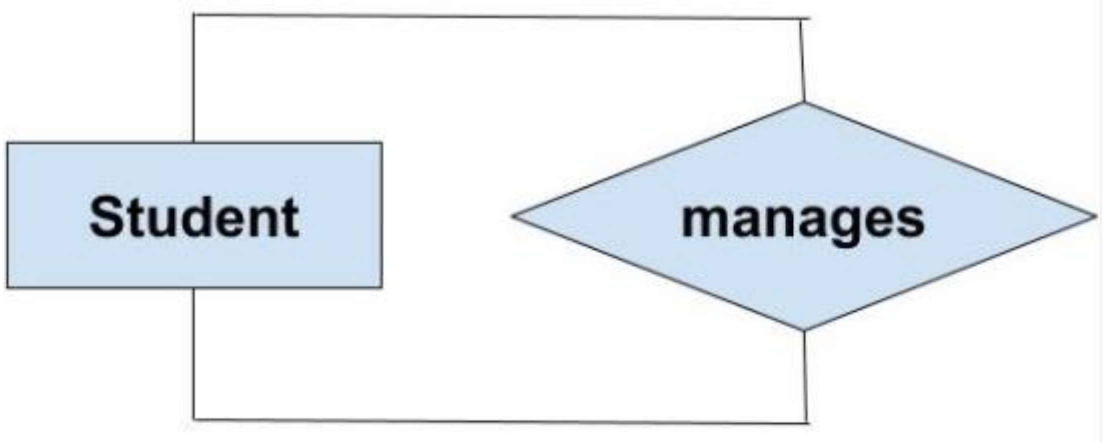


Degree of a Relationship Set

- The number of different entity sets participating in a relationship set is called the degree of a relationship set
- Different types of degree of a Relationship set are
  - Unary Relationship
  - Binary Relationship
  - n-ary Relationship

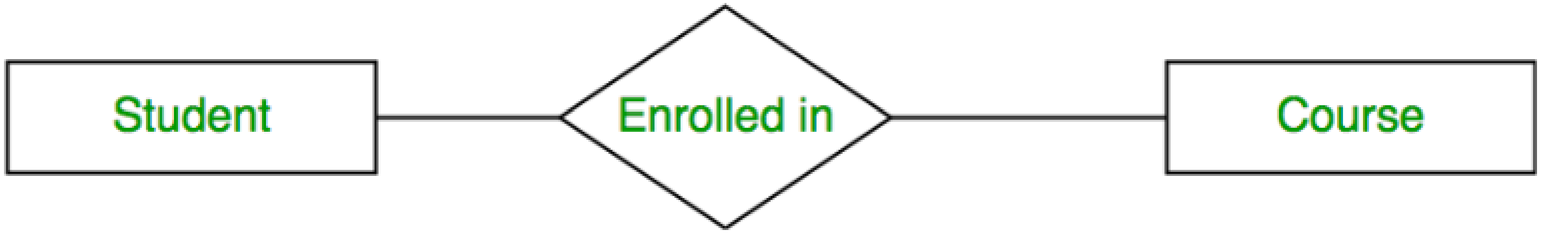
Unary Relationship

When there is only ONE entity set participating in a relation, the relationship is called a unary relationship



Binary Relationship

When there are TWO entities set participating in a relationship, the relationship is called a binary relationship. For example, a Student is enrolled in a Course.



n-ary Relationship

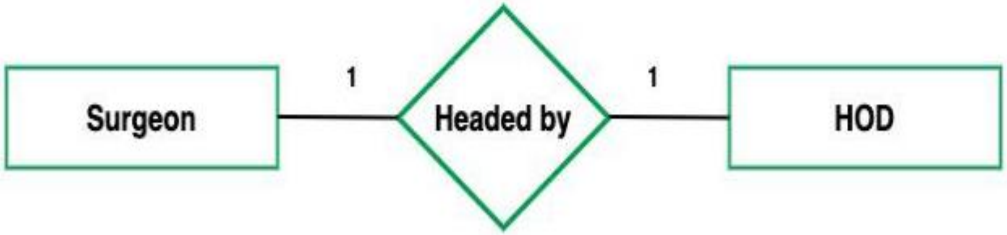
- When there are n entities set participating in a relation, the relationship is called an n-ary relationship.

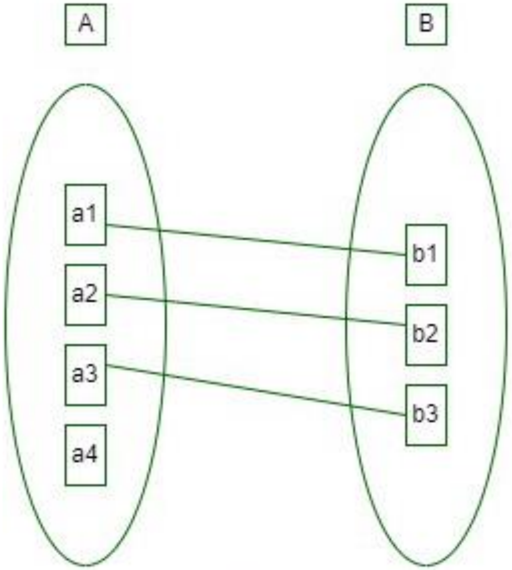
Cardinality

- The number of times an entity of an entity set participates in a relationship set is known as cardinality.
- Cardinality can be of different types
  - One-to-One
  - One-to-Many
  - Many-to-One
  - Many-to-Many

One-to-One

- When each entity in each entity set can take part only once in the relationship, the cardinality is one-to-one
- Example



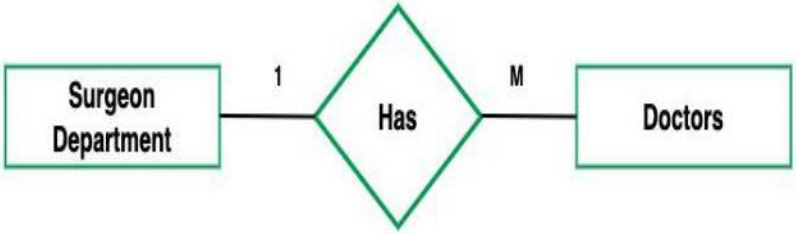


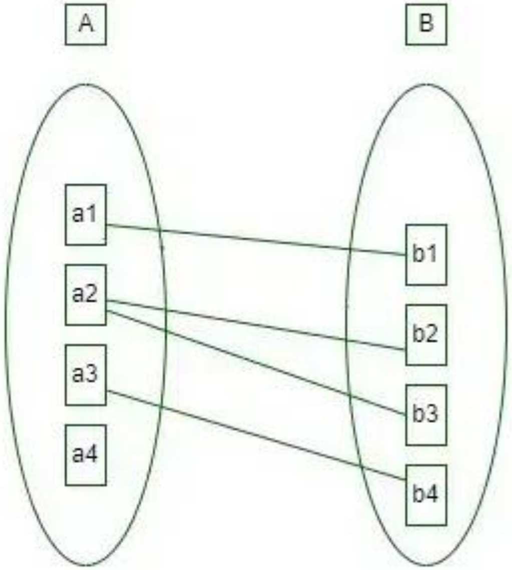
Set Representation of One-to-One

One-to-Many

- In one-to-many mapping as well where each entity can be related to more than one relationship and the total number of tables that can be used in this is 2
- Let us assume that one surgeon department can accommodate many doctors
- So the Cardinality will be 1 to M
- It means one department has many Doctors

One to Many (Contd..)



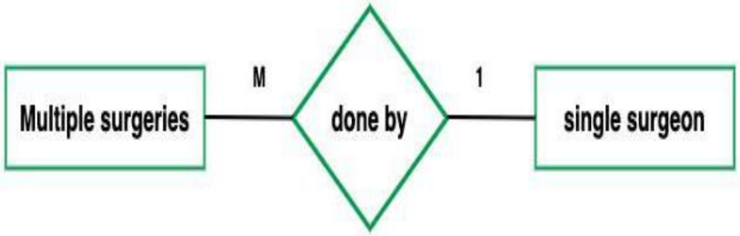


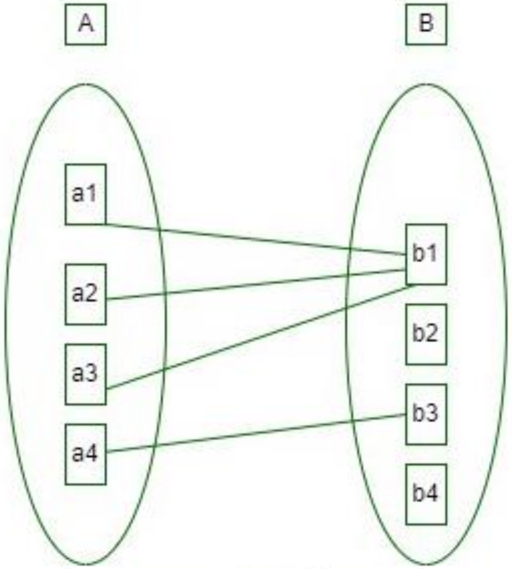
Set Representation of One-to-Many

Many-to-One

- When entities in one entity set can take part only once in the relationship set and entities in other entity sets can take part more than once in the relationship set, cardinality is many to one

Many to one (Contd..)





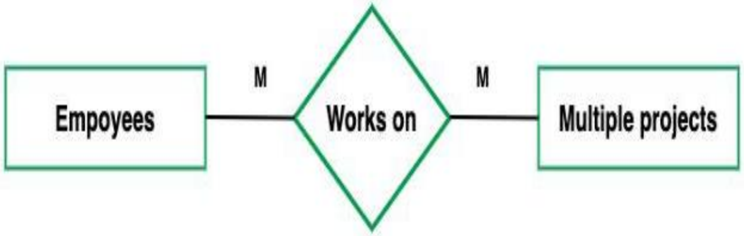
Set Representation of Many-to-One

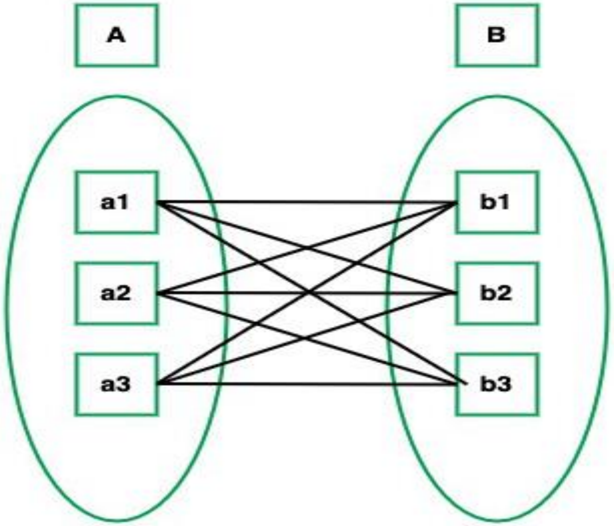
Many-to-Many

- When entities in all entity sets can take part more than once in the relationship cardinality is many to many

- For example, student S1 is enrolled in C1 and C3 and Course C3 is enrolled by S1, S3, and S4. So it is many-to-many relationships

Many to Many (Contd..)





Many-to-Many Set Representation

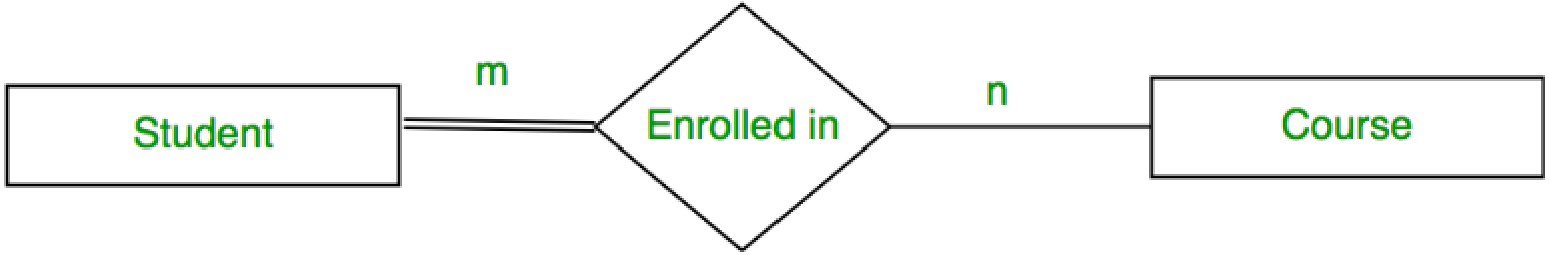
Participation Constraint

- Participation Constraint is applied to the entity participating in the relationship set
- **Total Participation**
- Each entity in the entity set must participate in the relationship
- If each student must enroll in a course, the participation of students will be total
- Total participation is shown by a double line in the ER diagram

Participation Constraint

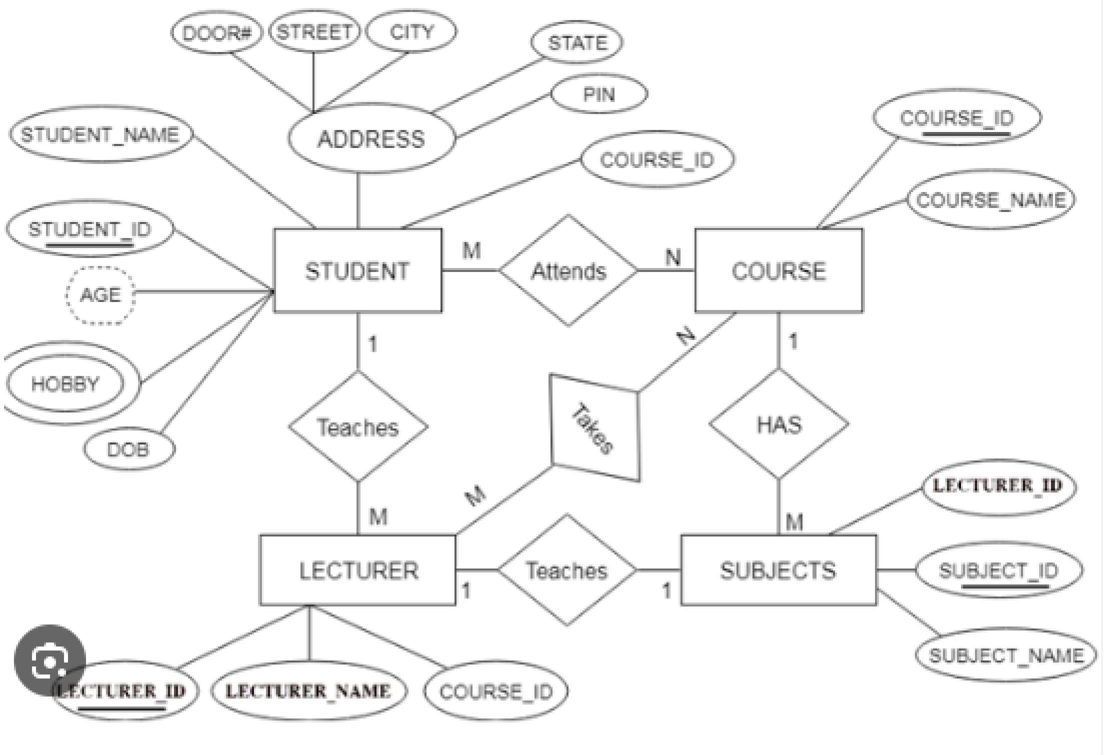
- **Partial Participation**
- The entity in the entity set may or may NOT participate in the relationship
- If some courses are not enrolled by any of the students, the participation in the course will be partial

Total and Partial participation

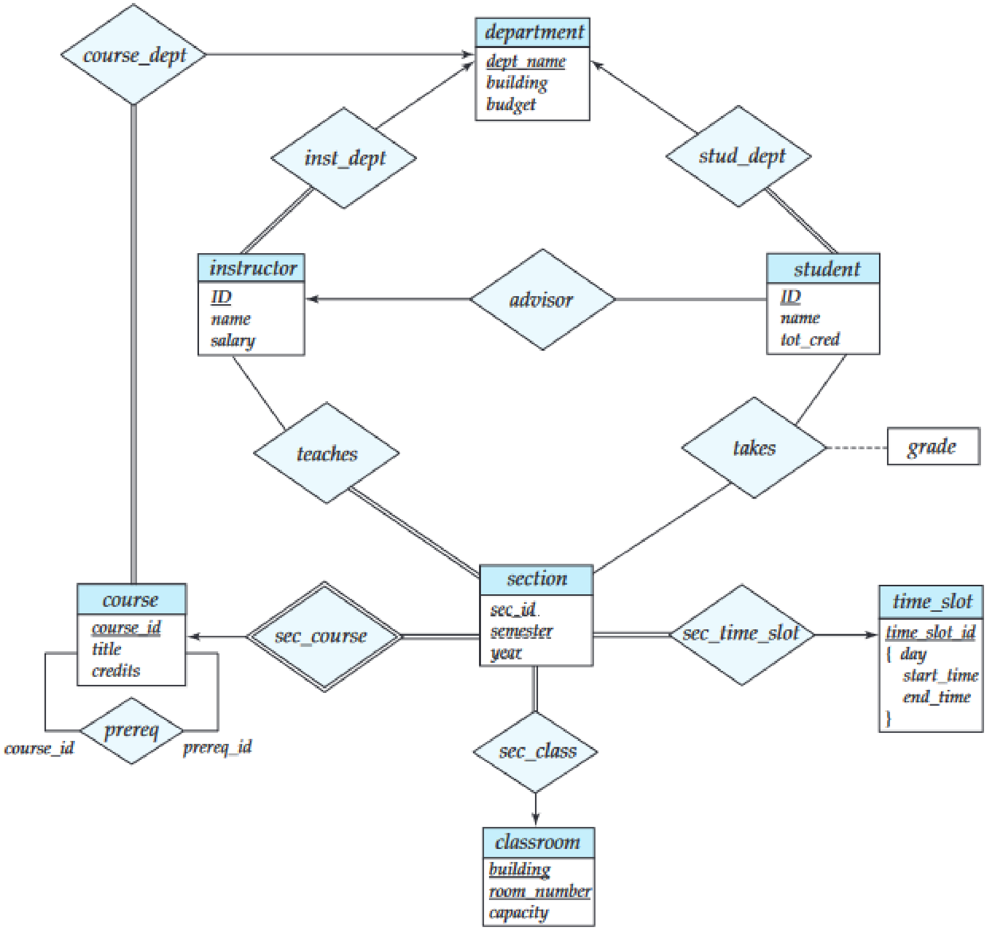


Total Participation and Partial Participation

Example


DOOR#
STREET
CITY
STATE
PIN
STUDENT_NAME
ADDRESS
COURSE_ID
COURSE_ID
COURSE_NAME
STUDENT_ID
AGE
HOBBY
DOB
STUDENT
M
Attends
N
COURSE
1
N
1
Teaches
Takes
HAS
M
M
M
LECTURER_ID
LECTURER
Teaches
SUBJECTS
SUBJECT_ID
1
1
SUBJECT_NAME
LECTURER_ID
LECTURER_NAME
COURSE_ID

Example


course_dept
department
dept_name
building
budget
inst_dept
stud_dept
instructor
ID
name
salary
advisor
student
ID
name
tot_cred
teaches
takes
grade
section
sec_id
semester
year
course
course_id
title
credits
sec_course
sec_time_slot
time_slot
time_slot_id
{ day
start_time
end_time
}
prereq
course_id
prereq_id
sec_class
classroom
building
room_number
capacity

Extended E-R Features

- Specialization
- Generalization
- Higher- and lower-level entity sets
- Attribute inheritance
- Aggregation

Specialization

- An entity set may include subgroupings of entities that are distinct in some way from other entities in the set.
- The process of designating subgroupings within an entity set is called specialization
- The specialization of person allows us to distinguish among person entities according to whether they correspond to employees or students
- The specialization relationship may also be referred to as a superclass-subclass relationship
- Example :The university could create two specializations of student, namely graduate and undergraduate
  - Overlapping specialization
  - Disjoint specialization

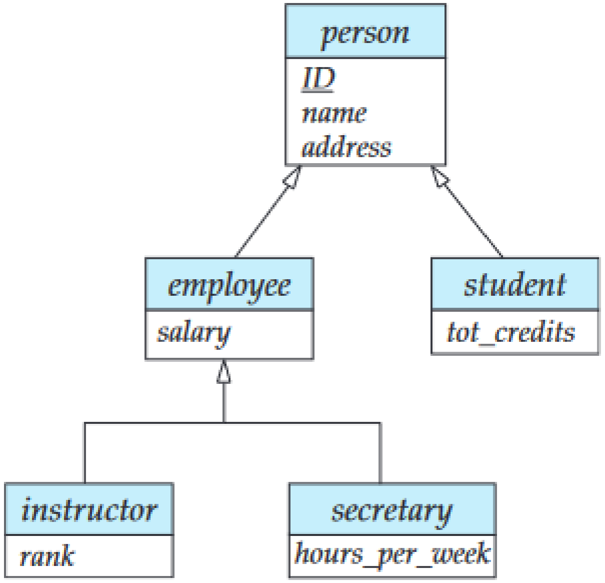
Specialization (Contd..)

- For an overlapping specialization (as is the case for student and employee as specializations of person), two separate arrows are used.
- For a disjoint specialization (as is the case for instructor and secretary as specializations of employee), a single arrow is used

Generalization

- Generalization proceeds from the recognition that a number of entity sets share some common features (namely, they are described by the same attributes and participate in the same relationship sets)

Specialization and generalization



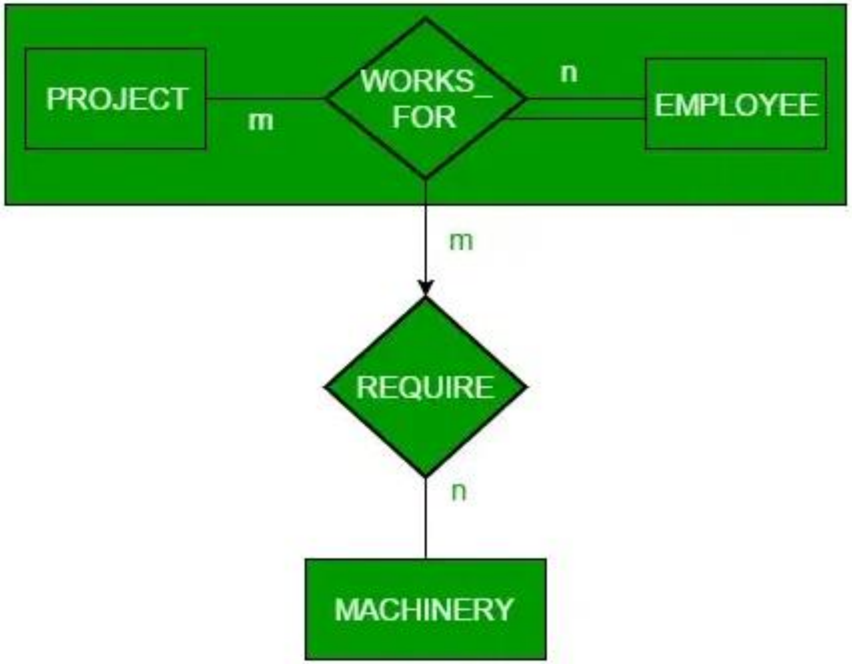
Attribute Inheritance

- Attribute inheritance: allows lower level entities to inherit the attributes of higher level entities and vice versa
- Example: Car entity is an inheritance of Vehicle entity ,So Car can acquire attributes of Vehicle
- Example:car can acquire Model attribute of Vehicle

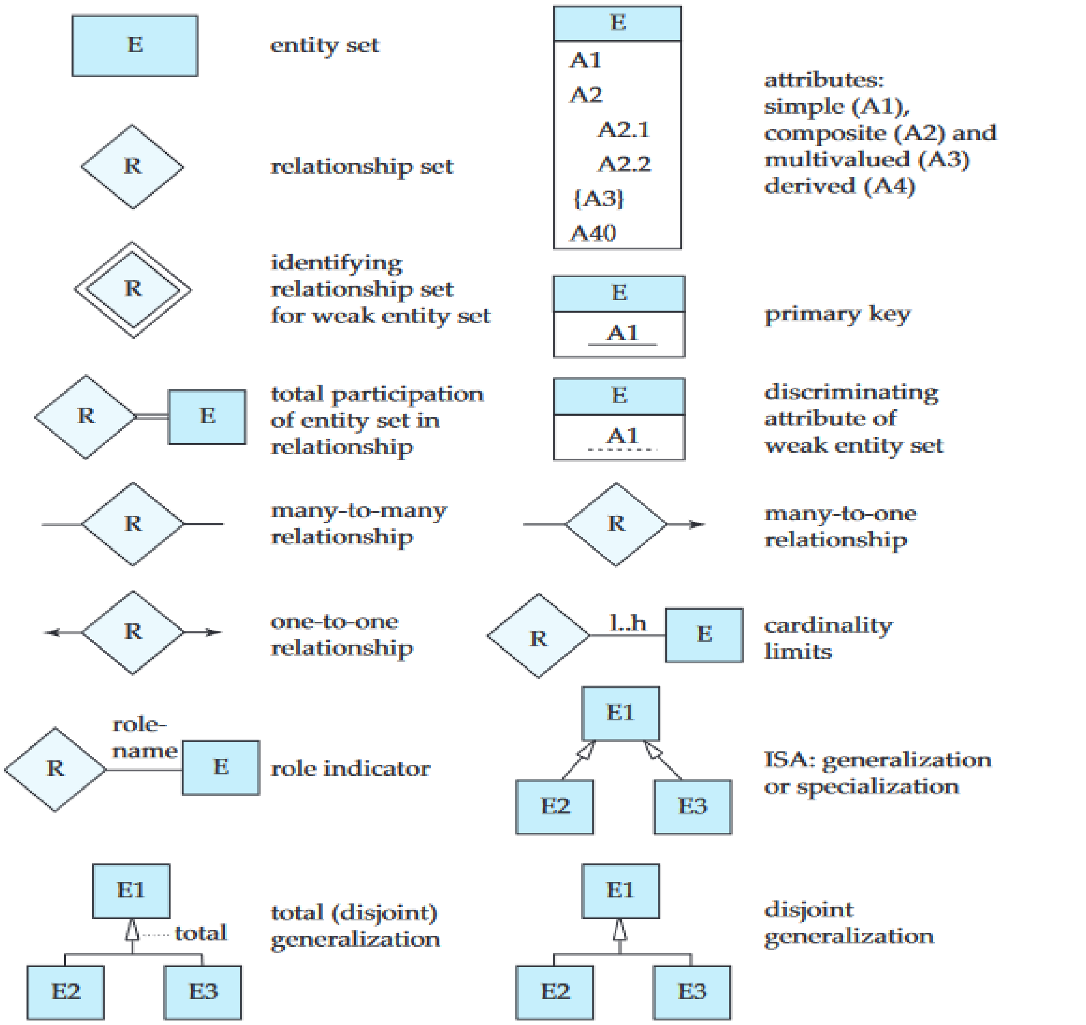
Aggregation

- An ER diagram is not capable of representing the relationship between an entity and a relationship which may be required in some scenarios
- In those cases, a relationship with its corresponding entities is aggregated into a higher-level entity
- Aggregation is an abstraction through which we can represent relationships as higher-level entity sets
- For Example, an Employee working on a project may require some machinery
- So, REQUIRE relationship is needed between the relationship WORKS_FOR and entity MACHINERY
- Using aggregation, WORKS_FOR relationship with its entities EMPLOYEE and PROJECT is aggregated into a single entity and relationship REQUIRE is created between the aggregated entity and MACHINERY.

Aggregation



Symbols used in ER notation



entity set

relationship set

identifying
relationship set
for weak entity set

total participation
of entity set in
relationship

many-to-many
relationship

one-to-one
relationship

role indicator

total (disjoint)
generalization

attributes:
simple (A1),
composite (A2) and
multivalued (A3)
derived (A4)

primary key

discriminating
attribute of
weak entity set

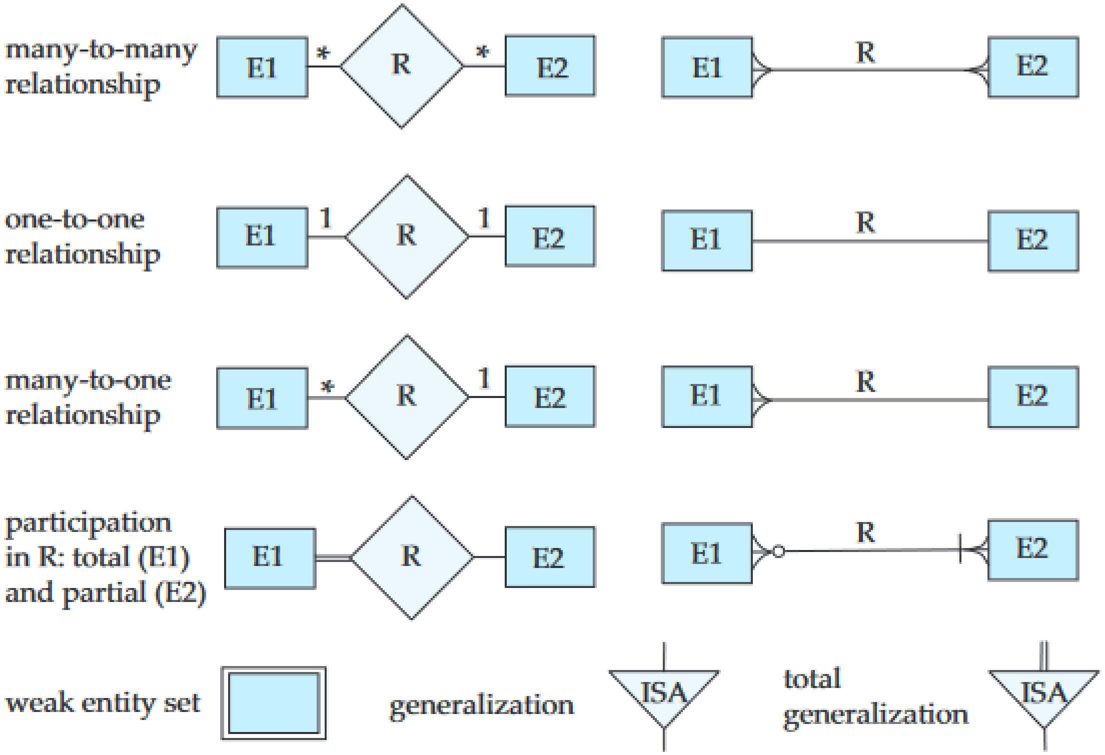
many-to-one
relationship

cardinality
limits

ISA: generalization
or specialization

disjoint
generalization

Alternative ER notations


many-to-many relationship
E1
*
R
*
E2
E1
R
E2
one-to-one relationship
E1
1
R
1
E2
E1
R
E2
many-to-one relationship
E1
*
R
1
E2
E1
R
E2
participation in R: total (E1) and partial (E2)
E1
R
E2
E1
R
E2
weak entity set
generalization
ISA
total generalization
ISA