DATA MODELING

ERD ⟷ CROW'S

S Chatterjee

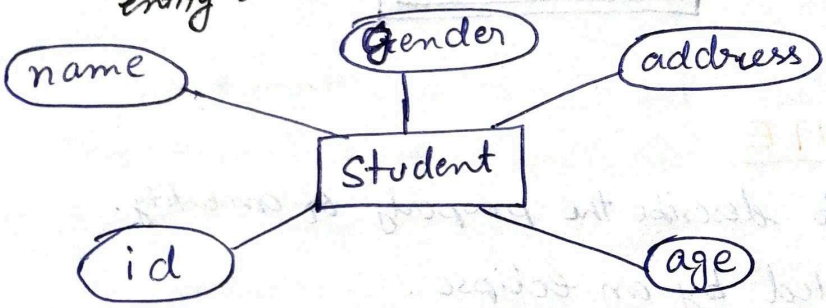
By
SIDDHARTH
CHATTERJEE

ER Model Concept

Entity Relationship is a high-level data model to define data elements and relationship for a specified system.

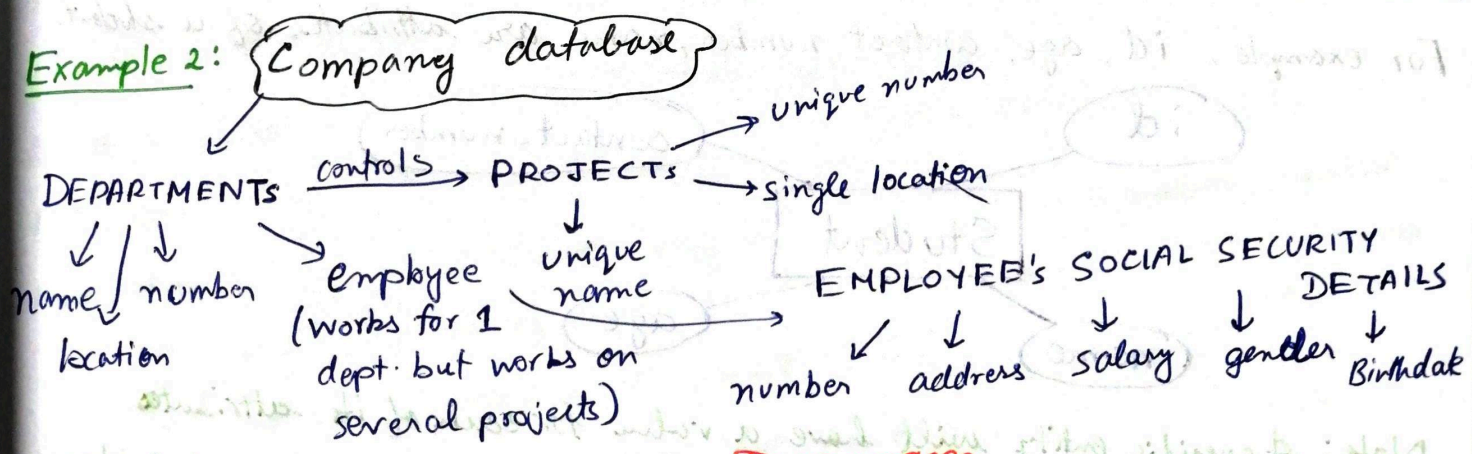
Develops a conceptual, simple and easy view of database

Example 1: Designing (school database) Student will be entity with attributes like address, name, id, age, gender, etc. Address can be another entity with attributes like city, streetname, pin code, etc.

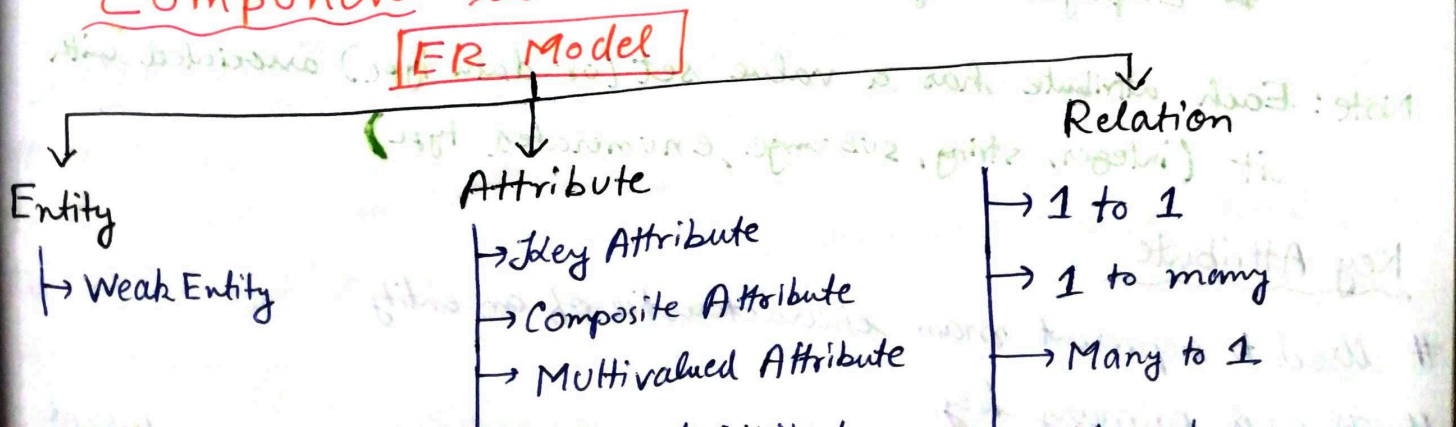


"ER Modeling is a Top-Down structure"

Example 2: (Company database)



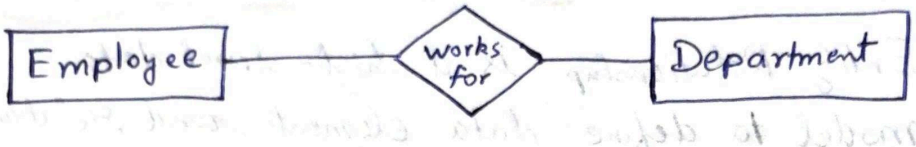
Component of ER Diagram



ENTITY

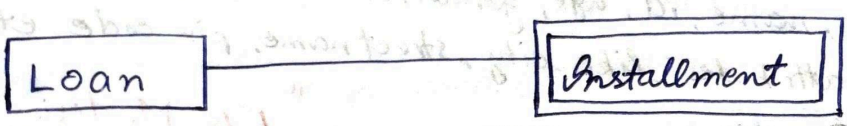
An entity may be an object, class, person or place.
In ER diagram, an entity can be represented by rectangles.

Consider an organisation as an example - manager, product, employee, department, etc can be taken as an entity.



Weak Entity

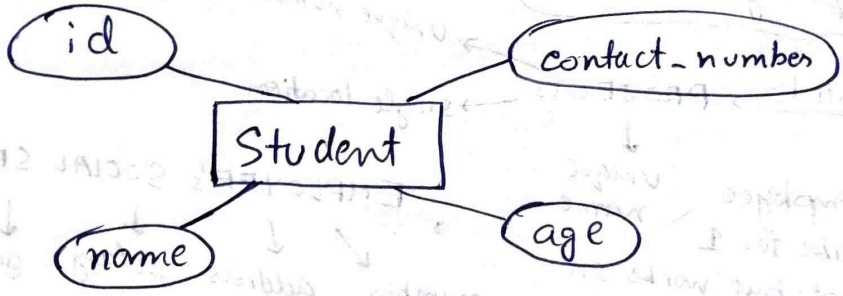
An entity that depends on another entity is called a weak entity
A weak entity doesn't contain any key attribute on its own.
A weak entity is represented by a double rectangle.



ATTRIBUTE

An attribute is used to describe the property of an entity.
An attribute is represented by an eclipse.

For example, id, age, contact number, name are attributes of a student.

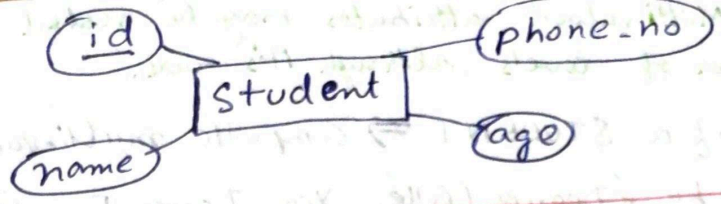


Note: A specific entity will have a value for each of its attributes
* Employee entity has name = 'John Smith', SSN='1234', Addres...

Note: Each attribute has a value set (or data type) associated with it (integer, string, subrange, enumerated type)

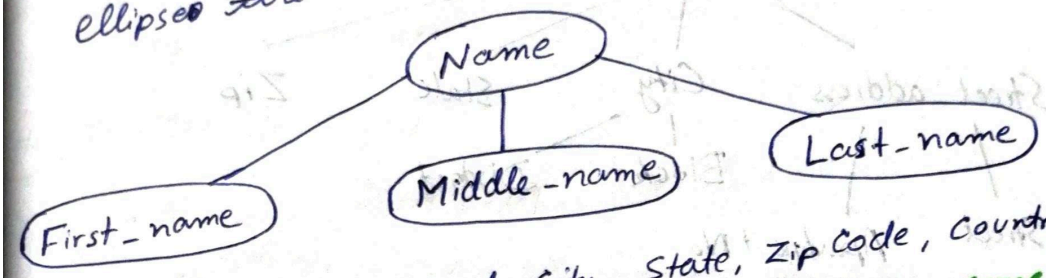
Key Attribute

Used to represent main characteristics of an entity
Has a primary key
A key attribute is represented by an ellipse with the text underlined



Composite Attribute

An attribute that is composed of many other attributes, or one can say several components
Composite attribute is represented by an ellipse and that is ellipses are connected with other ellipses.



Address (House, Street, City, State, Zip code, Country)

Note: Composition may form a hierarchy where some components are themselves composite.

Multivalued Attribute

An attribute can have more than 1 value; primarily because an entity may have multiple values for that attribute.
A multivalued attribute is represented by a double oval

For example: Colour of a CAR or Degrees of a STUDENT.

Note: A multi valued attribute may have lower and upper bounds to constrain the no. of values allowed for each individual attribute.

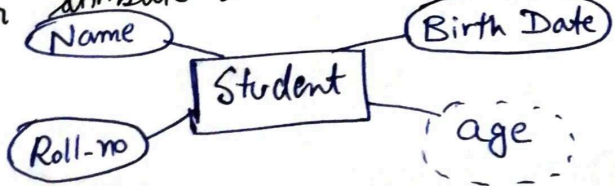
Denoted as {Color} or {Degrees}

Another Example: a student can have more than 1 phone number.

Derived Attribute is an attribute that

A derived attribute can be derived from other attribute
A derived attribute is represented by a dashed ellipse

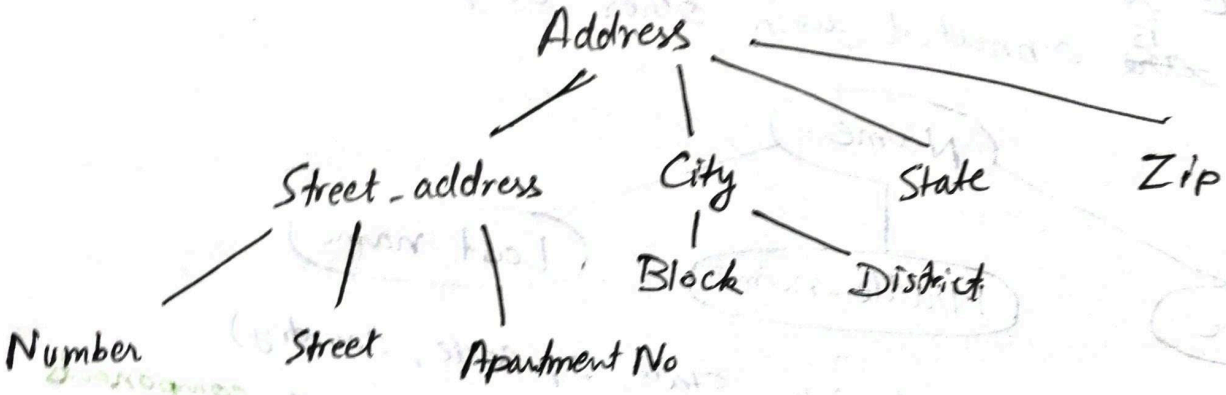
For example, a person's age changes over time and can be derived from another attribute like Date of Birth



Note: Composite & Multi valued attributes may be nested arbitrarily to any number of levels although it's rare.

Example:
- Degrees of a STUDENT ⇒ Composite multi-valued attribute denoted by {Degrees (College, Year, Degree, Field)}
- Multiple Degrees values can exist
- Each has 4 subcomponent attributes: College, Year, Degree, Field

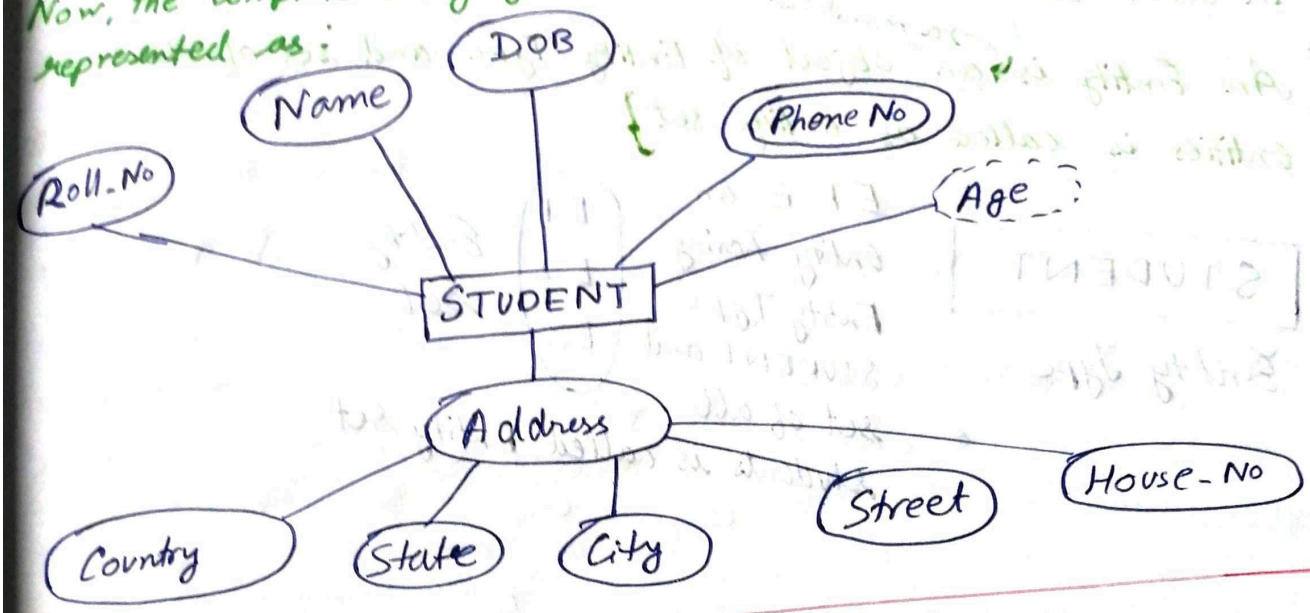
Composite Attribute Another Example



Stored v/s Derived Attribute

- Age attribute is derived attribute (determined from current date and the value of Birth-date attribute)
- Birth-date is a stored attribute

Now, the complete entity type STUDENT with its attributes can be represented as:



A Brief Overview of ENTITY, ENTITY SET & ENTITY TYPE

Basically; an entity is an object in the real world with an independent existence and can be differentiated from other objects.

Remember; an entity might be:
- An object with physical existence (lecturer, student, car)
- An object with conceptual existence (course, job, position)

ENTITY TYPE → Group of objects with the same properties that are identified by the enterprise as having an independent existence.

→ Entities with the same basic attributes are grouped or typed into an Entity Type

→ Collection of Entity having common attribute

→ Collection of similar entities.

ENTITY SET → Set of entities of same entity type = Entity Set

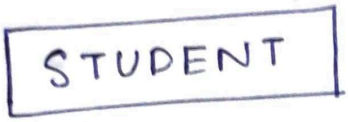
→ Each entity type will have a collection of entities stored in the database called the entity set

→ Current state of the entities of that type that are stored in the database.

ENTITY OCCURENCE → A uniquely identifiable object of an entity type

So, in short conclusion:

{ An Entity is an object of Entity Type (a member/instance) and set of all entities is called as entity set }



Entity Type

E1 is an entity having Entity Type STUDENT and set of all students is called Entity Set

Entity Set

RELATIONSHIP

A relationship is used to describe the relation between entities.

A relationship is represented by a diamond / rhombus.

A relationship relates entities with a specific meaning

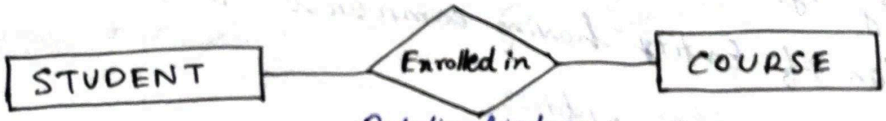
For example: EMPLOYEE John Smith works on Product Z PROJECT

EMPLOYEE Franklin Worg manages the Research DEPARTMENT

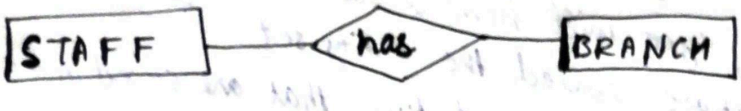
RELATIONSHIP TYPE

A relationship type represents association between 1 or more participating entity types.

Each relationship type is given a name that describes its function.



Relationship type in diamond connecting entities with lines.



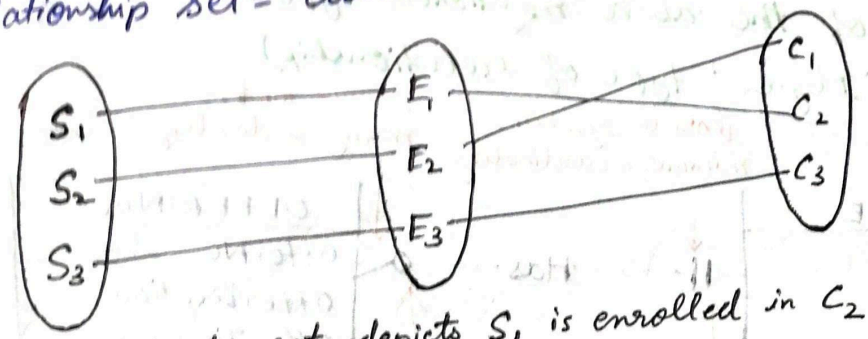
Relationships of same type are grouped or typed into a relationship type.

Basically, the schema description of a relationship

Also, identifies certain relationship

RELATIONSHIP SET

A set of relationships of same type is called as relationship set
A relationship set is the 'current set' of relationship instances represented in the database
Relationship set = 'current state' of relationship type



This relationship set depicts S_1 is enrolled in C_2, S_2 is enrolled in C_1 and S_3 is enrolled in C_3

DEGREE OF A RELATIONSHIP SET

The number of different entity types participating in a relationship set is called 'Degree of a relationship set'.

CROW'S FOOT NOTATION

Represents entities as boxes and relationships as lines between the boxes.
Different shapes at the end of these lines represent the relative cardinality of the relationship.

Acc. to Patrycja Dybka:

"it looks like a fork (inverted arrow) because it does not imply directionality or a physical access path rather it is visually intuitive, showing manyness"

[X] ------< [Y]

{Here an individual x can relate to multiple Ys (and each Y relates to at most one X}

NOTE: Relationships have 2 indicators, shown on both sides of the line.

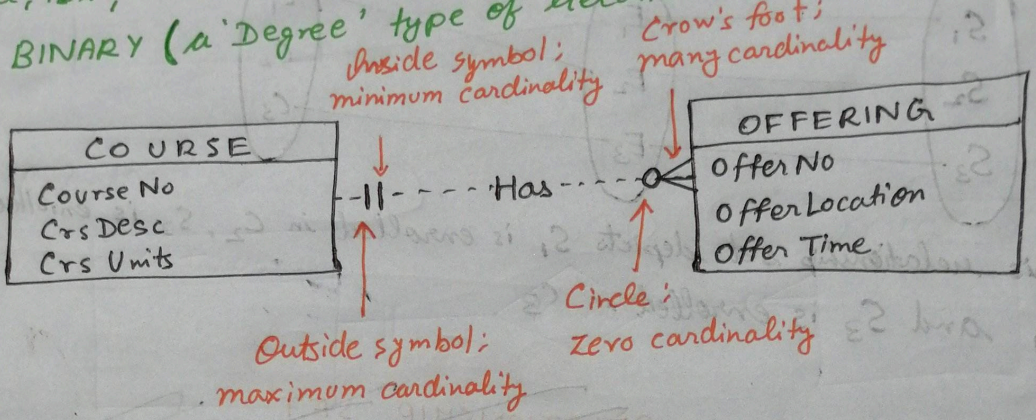
★ 3 symbols are used to represent cardinality

- ring → "0"
- dash → "1"
- Crow's feet → "many" or "infinite"

These symbols are used in pairs to represent the 4 types of cardinality that an entity may have in a relationship.

Note You must be wondering why is there no mention of degree of a relationship type, yet!

Well, tbh, all of the above mentioned type of relationship is BINARY (a 'Degree' type of relationship)



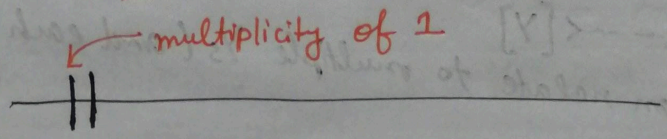
The "OUTSIDE SYMBOL" (OUTER COMPONENT) (Maximum Cardinality)

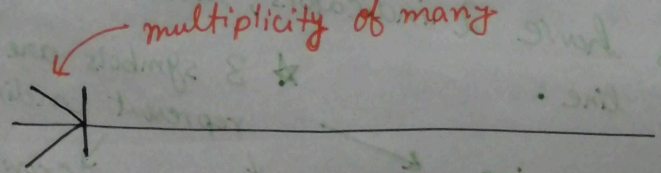
Outer/outside with respect to relationship verb "Has" से काफी दूर है!

This is the 1st indicator from left to right

Often called "multiplicity" this refers to the maximum no of times that an instance of 1 entity can be associated with instances in the related entity.

It can be 1 or many





The "INSIDE
Cardinali

Inner/ Ins
"Has" 2

This is the

Refers to th
related to

It can be

This accord
mandatory

Note The
ALW
Symbol
F

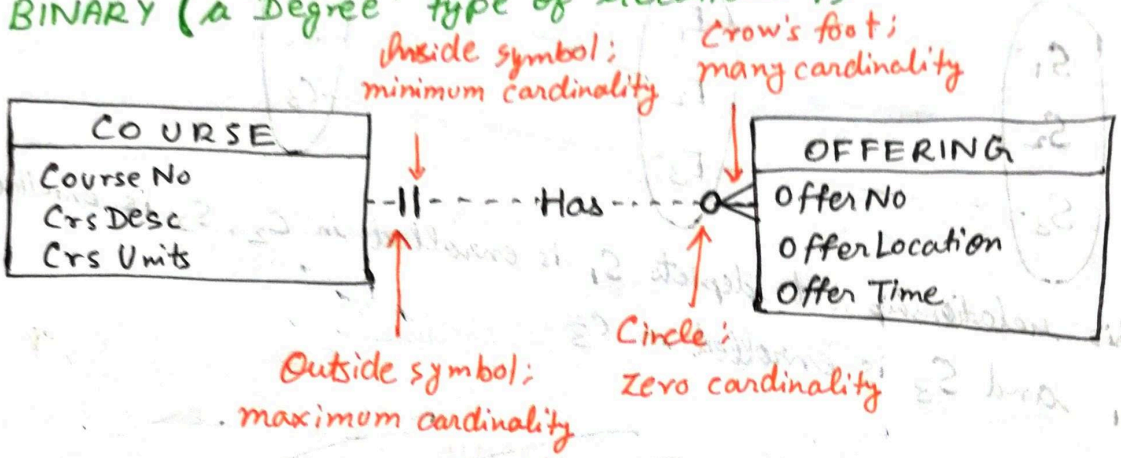
Therefore in
edges (4 ty
(Min)
① Zero or

② (Min)
One or

These symbols are used in pairs to represent the 4 types of cardinality that an entity may have in a relationship.

Note You must be wondering Why is there no mention of degree of a relationship type, yet!

Well, tbh, all of the above mentioned type of relationship is BINARY (a 'Degree' type of relationship)



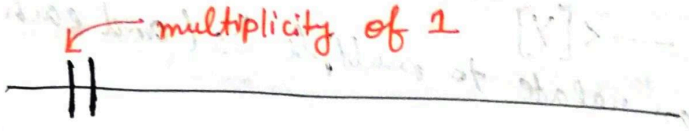
The "OUTSIDE SYMBOL" (OUTER COMPONENT) (Maximum Cardinality)

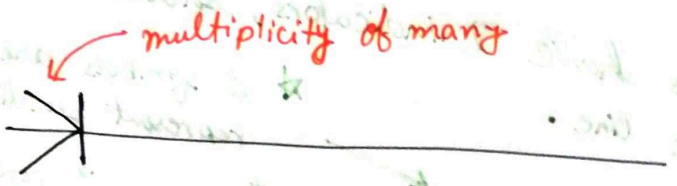
Outer/outside with respect to relationship verb "Has" रेखा से काफी दूर है!

This is the 1st indicator from left to right

Often called "multiplicity" this refers to the maximum no. of times that an instance of 1 entity can be associated with instances in the related entity.

It can be 1 or many





The "INSIDE SYMBOL" (INNER COMPONENT) / Minimum Cardinality

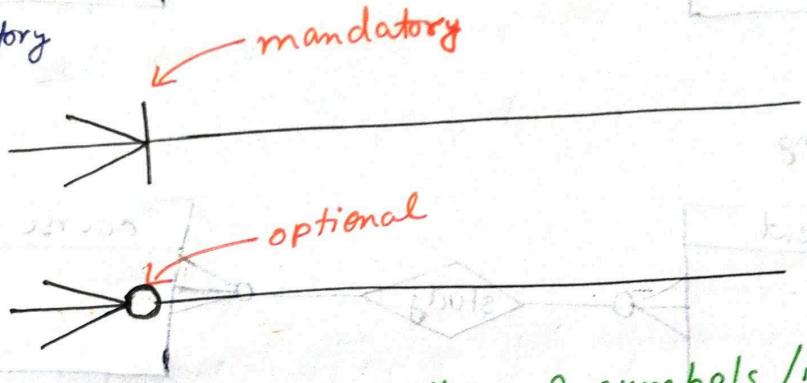
Inner / Inside with respect to relationship verb "Has" से काफी ~~दूर~~ पास है !

This is the 2nd indicator from left to right

Refers to the minimum no. of times 1 instance can be related to others

It can be 0 or 1

This accordingly, describes the relationship as optional or mandatory

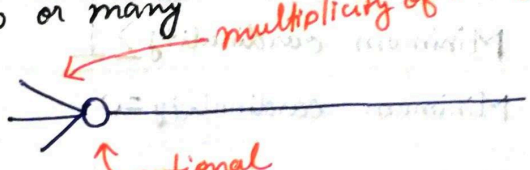


Note The combination of these 2 symbols / indicators IS ALWAYS IN A SPECIFIC ORDER !!

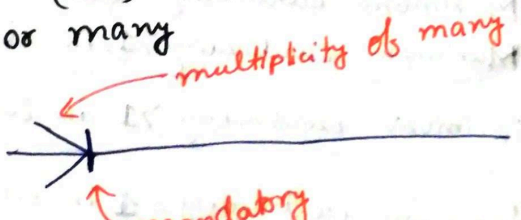
Symbol of multiplicity (FIRST) then whether Relationship is mandatory or optional (SECOND)

Therefore in reference to # Hf on previous page, there are 4 possible edges (4 types of cardinality) to a relationship

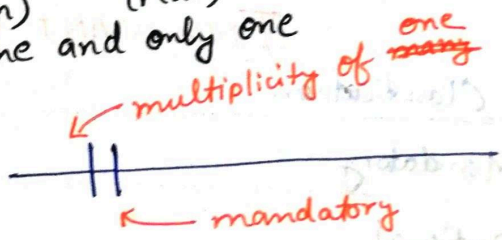
① (Min) Zero (Max) or many



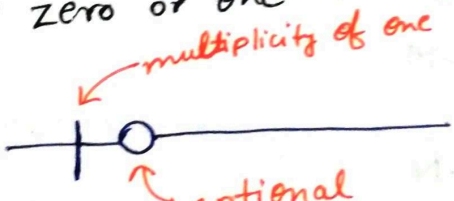
② (Min) One (Max) or many



③ (Min) One and only one (Max)

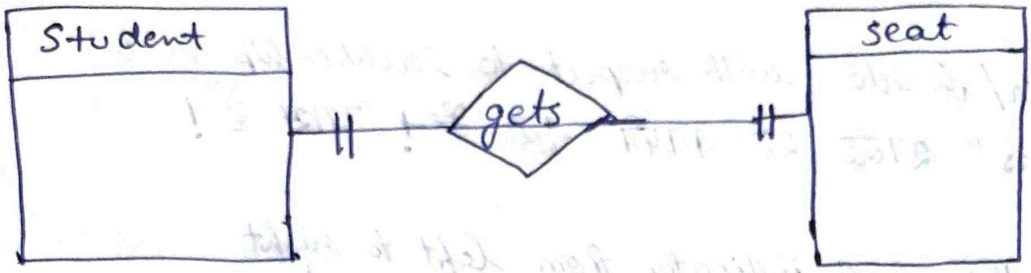


④ (Min) Zero or one (Max)

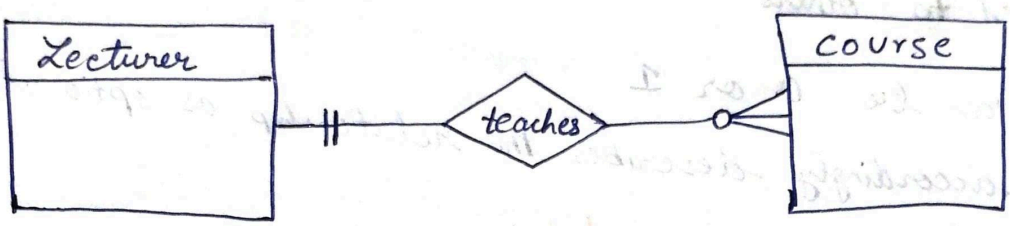


A binary relationship degree will make it readable as :-

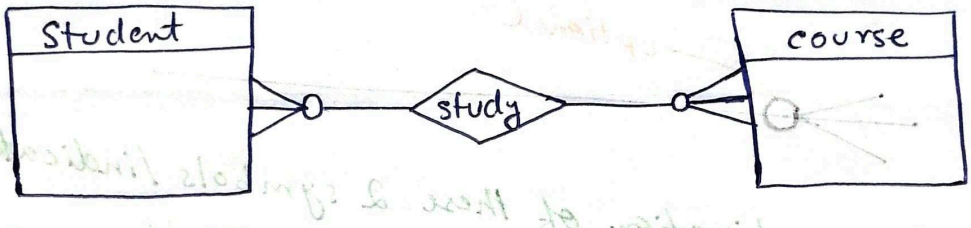
① One-to-one



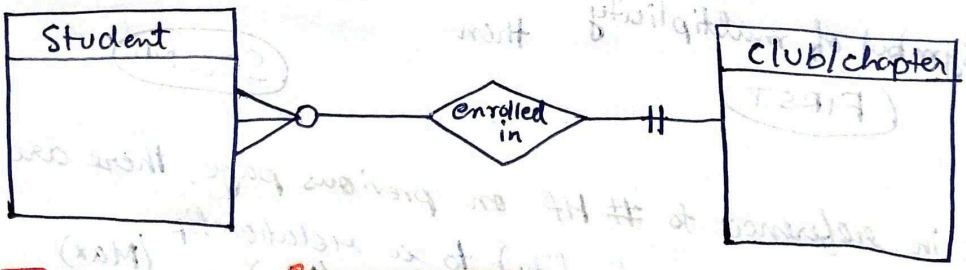
② One-to-many



③ Many-to-many



④ Many-to-one



IMPORTANT CARDINALITIES

Classification	Cardinality Restrictions
Mandatory	Minimum cardinality ≥ 1
Optional	Minimum cardinality $= 0$
Functional/Single valued	Minimum cardinality $= 1$
1-M	Maximum cardinality $= 1$ in one dir^n Maximum cardinality > 1 in other dir^n
M-N	Maximum cardinality > 1 in both dir^n
1-1	Maximum cardinality $= 1$ in both dir^n

Degree of a Relationship

Number of participating entities in a relationship type is referred to as degree of that particular relationship type

This can be reconstructed to : No. of participating entity types / sets in a relationship set is called as degree of the relationship set

3 types
- Unary → 1
- Binary → 2
- n-ary → n

CONCLUSION

- ERD notation has never been effectively standardized.
- And it is doubtful that standards will ever emerge
- No vendor has enough market power to force standardization :(