Chapter 4

Structuring System Requirements: Conceptual Data Modeling

Contents:

4. STRUCTURING SYSTEM REQUIREMENTS: Conceptual Data Modeling (4hrs)

a. Conceptual Model
b. Introduction to ER Model
c. Conceptual data modeling and ER Model
d. Role of CASE in conceptual data modeling

- We learnt about process modeling that analyzes flow of data between various processes and data stores.
- Similarly logic modeling concentrates on the logics involved in different processes .
- But non of these modeling techniques focused on the data.

4.1 Conceptual Data modeling

- A conceptual data model is a representation of organizational data.
- Focuses on the definition, structure, or relationship among the data.
- The purpose of a conceptual data model is to show as many rules about the meaning and interrelationships among data as possible, independent of any database management system or other implementation considerations.
- Entity-relationship (E-R) data models are commonly used diagrams that show how data are organized in an information system.

Information gathering for conceptual data model:

- Joint Application Design (JAD) :both user and technician
- Interviews and questionnaires.
- Data stored in repositories that may be manual or computerized.

Why conceptual data modeling?

- Characteristics of data captured from conceptual data modeling are crucial in design of databases, programs, computer screens, and printed documents.
  - E.g.: Facts such as, a data element is numeric, one product can have only one unique product id, etc. can ensure data integrity.

- Data are more complex than process as the systems are data-sensitive and require extracting data from various resources.

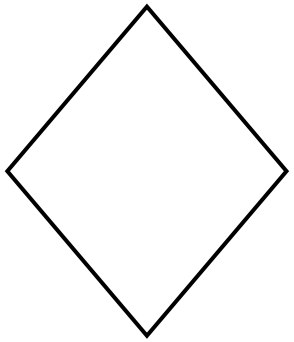
- Finally, structural information about data is essential to generate programs automatically.

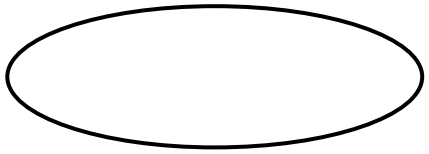
4.2 ER model

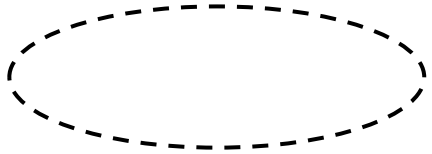
- An Entity-Relationship Model (ERM) is a detailed, conceptual and abstract representation of the data for an organization or for a business area.
- **Entity-Relationship Diagram (ERD)** is a graphical representation of a Entity-Relationship Model.
- ERD is expressed in terms of:
    - Entities
    - Attributes
    - Relationships/Associations
- The purpose of an ERD is to capture the richest possible understanding of the meaning of data necessary for an information system or organization.
- An ERD is a design or blueprint of a database that can later be implemented as a database.
- The E-R diagram is a model of entities in the business environment, the relationships or associations among those entities, and the attributes or properties of both the entities and their relationships.

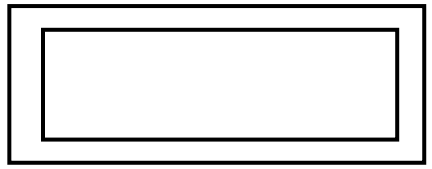
Symbols used in ER Diagram

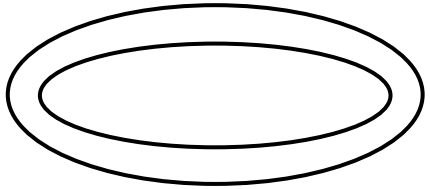
1.  Entity

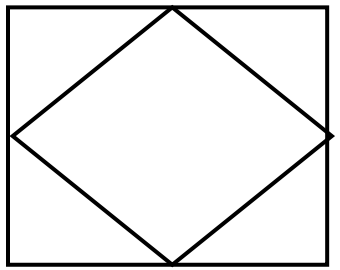
2.  Relationship

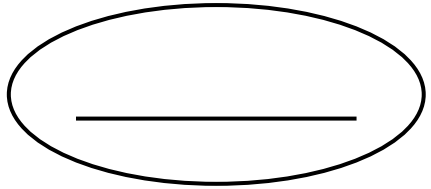
3.  Attribute

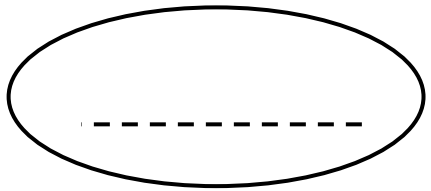
4.  Derived attribute

5.  Weak entity

6.  → Multivalued Attribute

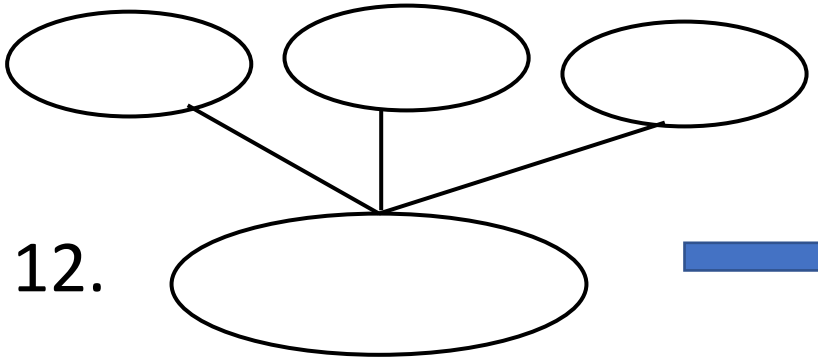
7.  → Associative Entity

8.  → Key Attribute

9.  → Foreign Key Attribute

10. _____ → Links attribute to entity set or entity set to relationships.

11. ═══ → Represents total participation of entity.



12. → A composite attribute

Components of ER Diagram

- An ER diagram has three main components:
    1. Entity
    2. Attribute
    3. Relationship

1.Entity

- An entity is a person, place, object, event, or concept in the user environment about which the organization wishes to maintain data.
- It has its own identity, which distinguishes it from every other entity.
- Symbol: Rectangle

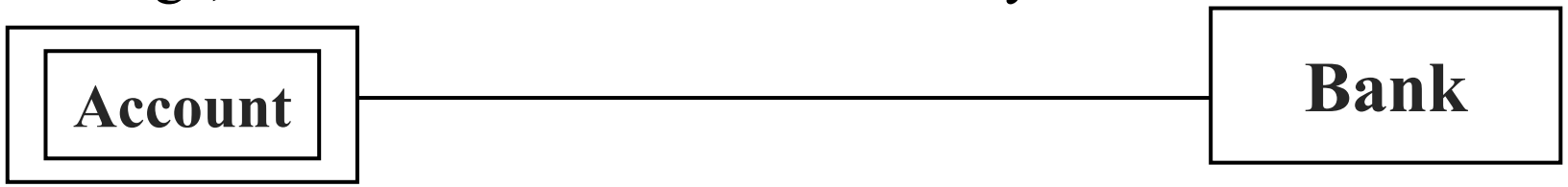
- Examples :
  - Person: Employee, Student , Patient , etc.
  - Object: Machine , Building , Automobile, etc.
  - Place: State , Region , Country , Branch , Location , etc.
  - Event: Registration , Renewal , etc.
  - Concept: Project , Course , Work Centre , Account , etc.

Student

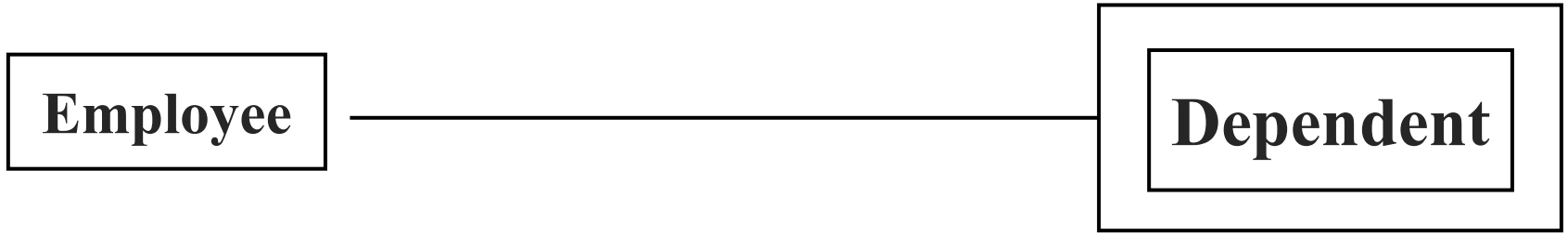
Account

Weak Entity

- Also called as dependent entity.
- It cannot be uniquely identified by its own attributes and relies on the relationship with other entity .
- Represented by a double rectangle.
- Examples:
  - a bank account cannot be uniquely identified without knowing the bank to which the account belongs, so bank account is a weak entity.

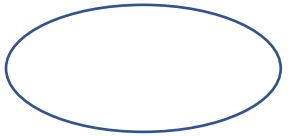


  - An employee has dependents.



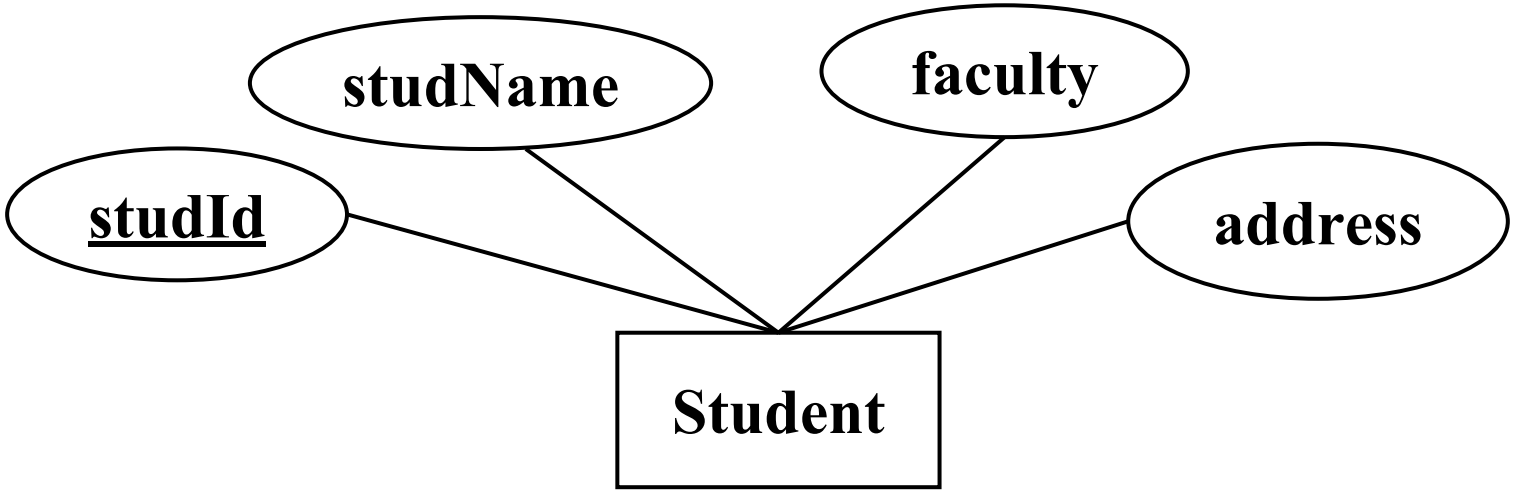
2.Attribute

- An attribute describes the property of an entity.
- It is represented by Oval in an ER diagram.



- Example:

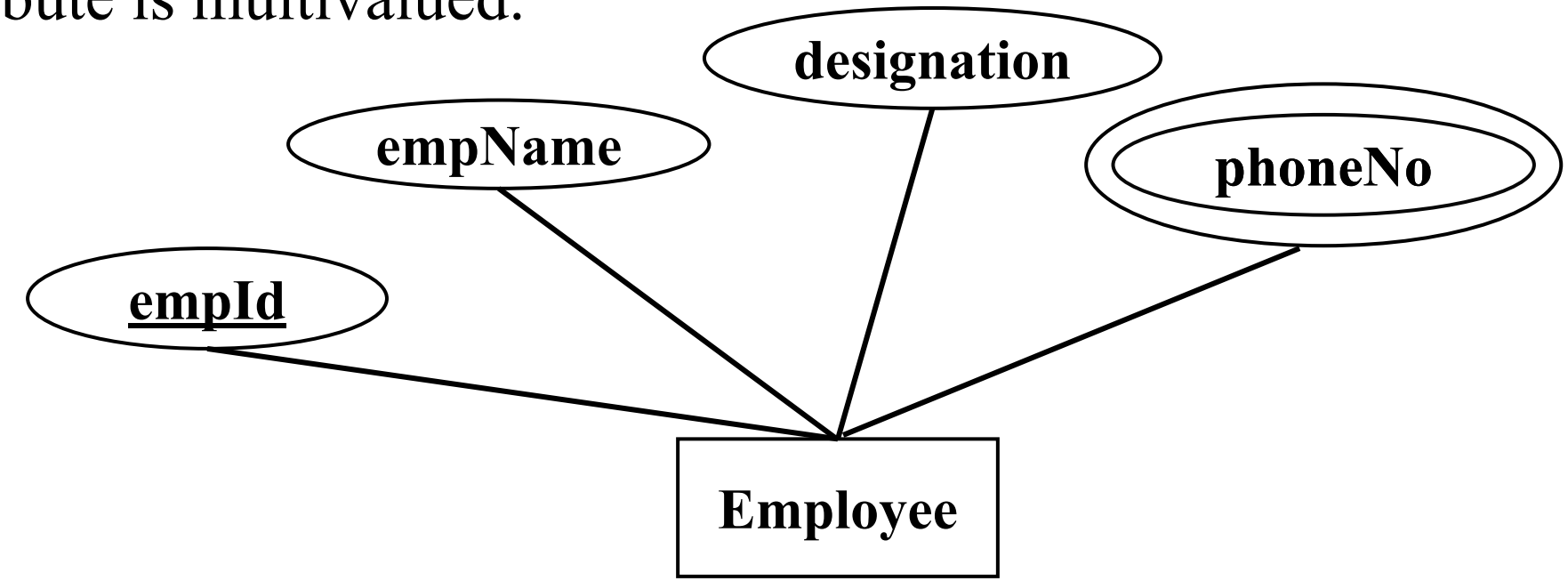
Student(<u>stdId</u> , stdName , faculty , address)



- There are four types of attributes:
    a. Multivalued attribute
    b. Derived attribute
    c. Key attribute
    d. Composite attribute

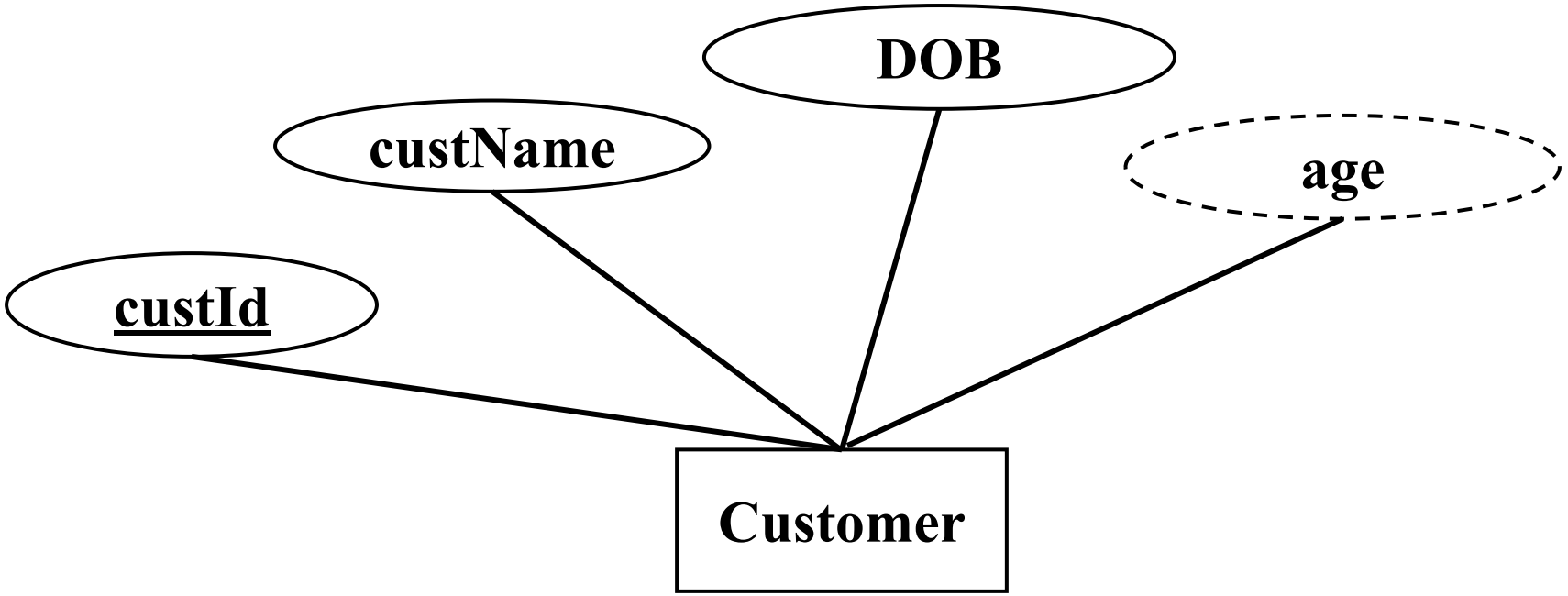
a)Multivalued attribute

- Can hold multiple values .
- Represented by **double ovals** in an ER Diagram.
- Example :
- An employee can have more than one phone numbers, so the phone number attribute is multivalued.



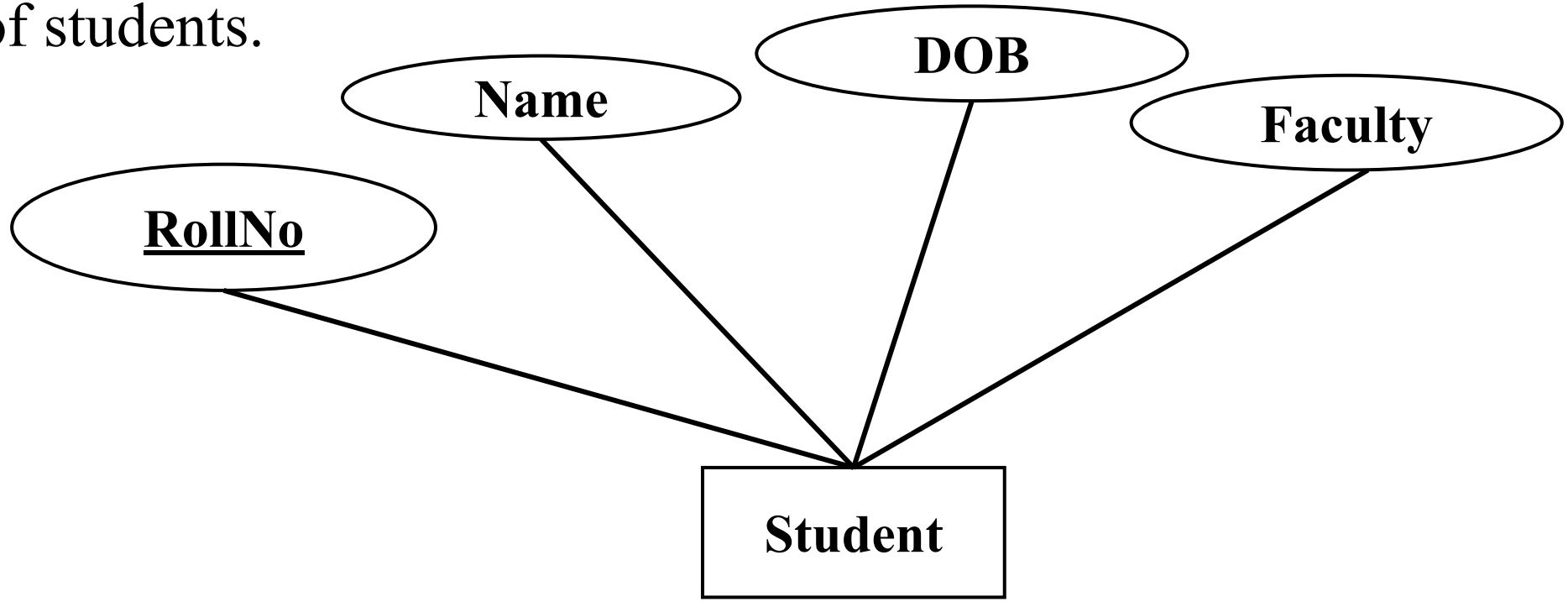
b)Derived attribute

- One whose value is dynamic and derived from another attribute.
- Represented by **dashed oval** in an ER Diagram.
- For example
  - Customer's age is a derived attribute as it changes over time and can be derived from another attribute (Date of birth).



c)Key /Identifier attribute

- An attribute or group of attributes that can uniquely identifies each entity instance.
- Represented by oval same as other attributes however the **text of key attribute is underlined**.
- For example, student roll number can uniquely identify a student from a set of students.

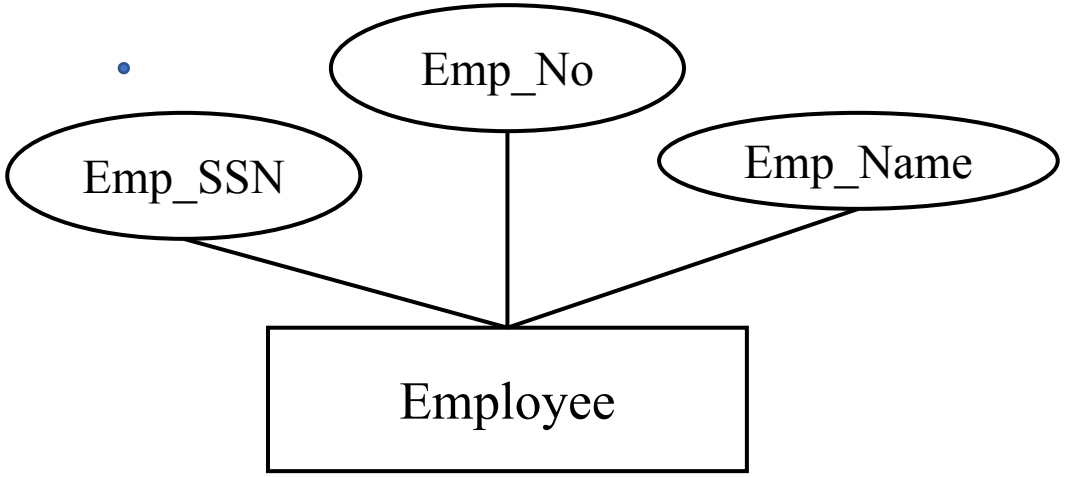


- **Types of key attributes:**
  - Super key
  - Candidate key
  - Composite/compound/concatenated key
  - Primary key
  - Alternate key
  - Foreign key

Assignment

Super key

- A super key is a set of one or more attributes (columns), which can uniquely identify a row in a table(entity).
- Example: Employee(Emp_SSN , Emp_Number , Emp_Name)

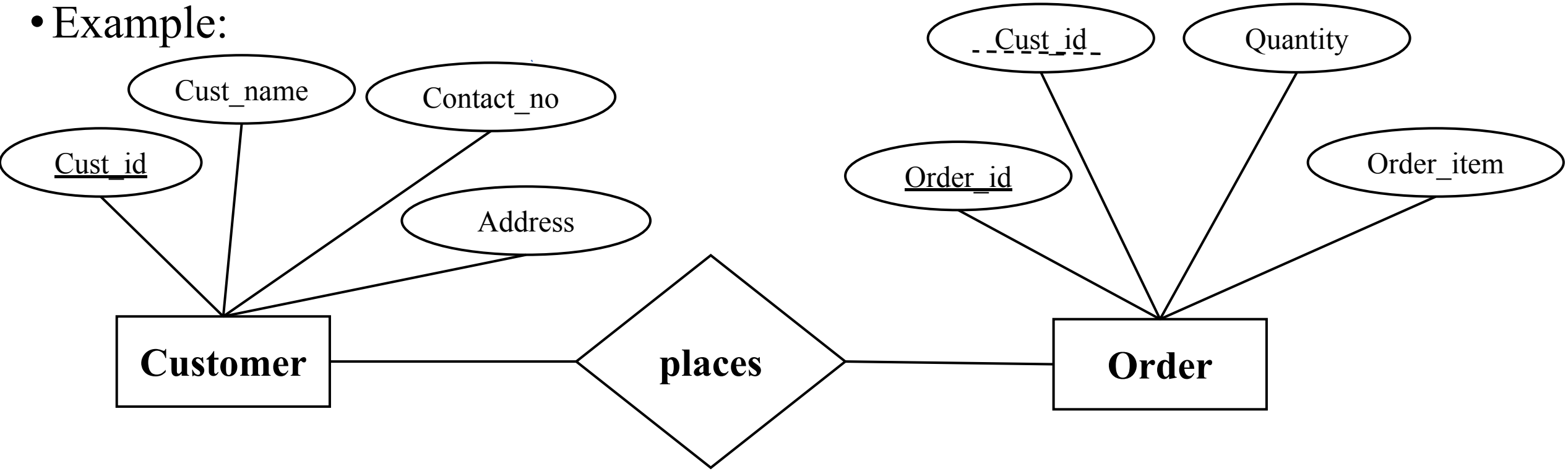


Emp_SSN	Emp_NO	Emp_Name
123456789	226	John
999999321	227	Steve
888997212	228	Robert
777778888	229	Ebhaan

- In above table ,sets of super key are :
  - { Emp_SSN }
  - { Emp_NO}
  - { Emp_SSN , Emp_NO}
  - { Emp_SSN , Emp_Name }
  - { Emp_SSN , Emp_NO , Emp_Name }
  - { Emp_NO , Emp_Name }

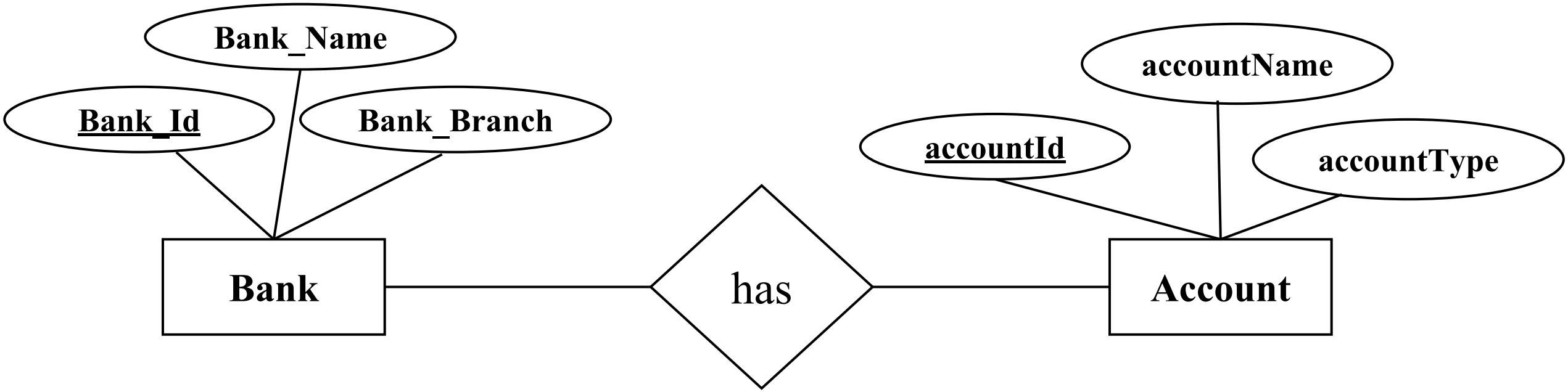
Foreign key

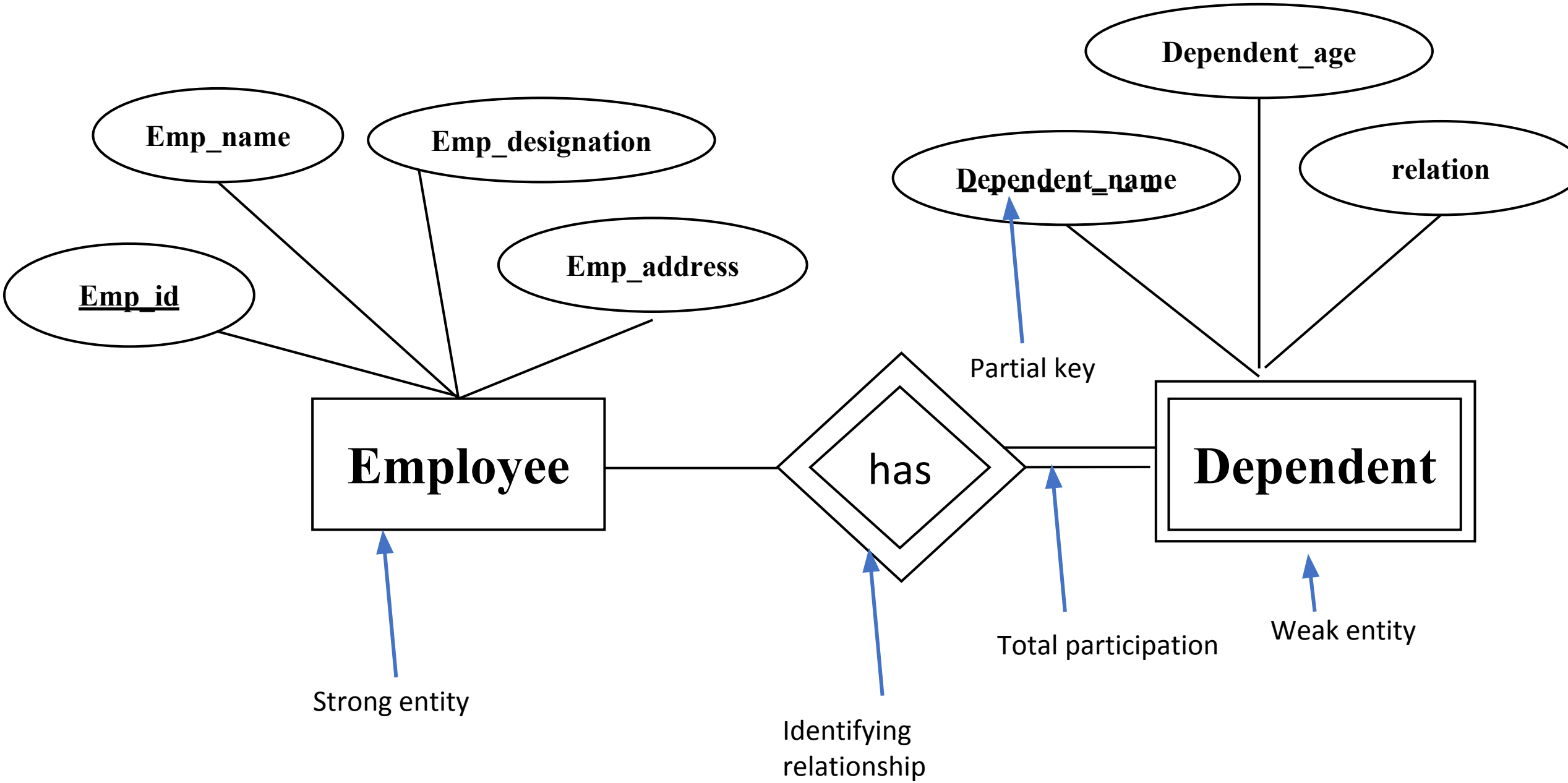
- A key used to link two entities/tables together.
- Sometimes also called as a referencing key.
- Foreign key is a primary key of one entity that is contributed to another entity to identify specific related entity instances of relationship.
- Example:



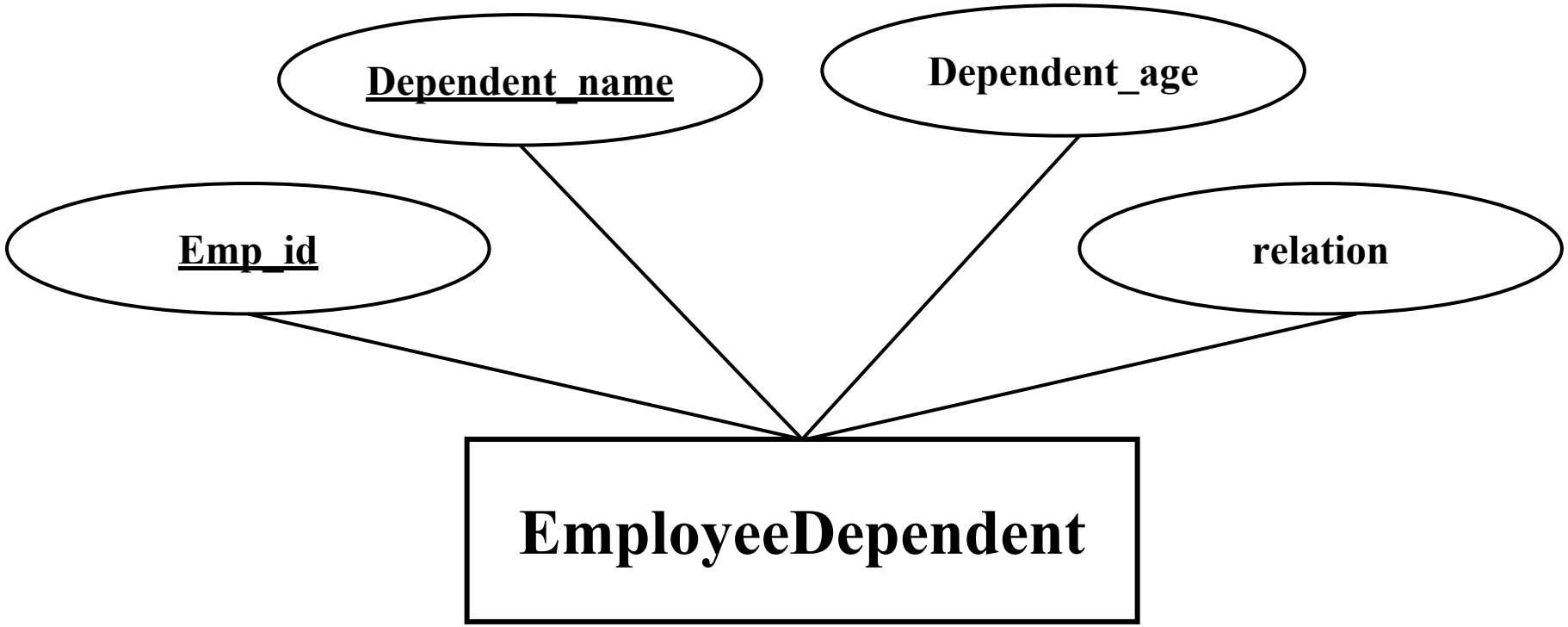
3.Relationship

- Relationship defines how entities are related to each other.
- An association between instances of one or more entities .
- Represented by **diamond shape** in ER Diagram.
- All entities participating in relationship are connected by a line.
- Example:





Note: Weak relation is represented using double diamond shape.


Dependent_name
Dependent_age
Emp_id
relation
EmployeeDependent

Cardinalities in relationship

- A measure of number of instances of an entity that can be associated with the instances of another entity.
- Kinds of cardinalities:
    I. Mandatory cardinalities
    II. One optional one mandatory cardinalities
    III. Optional cardinalities

I. Mandatory cardinalities

- One patient has at least one or more patient history. Or
- One patient history is associated to only one patient.

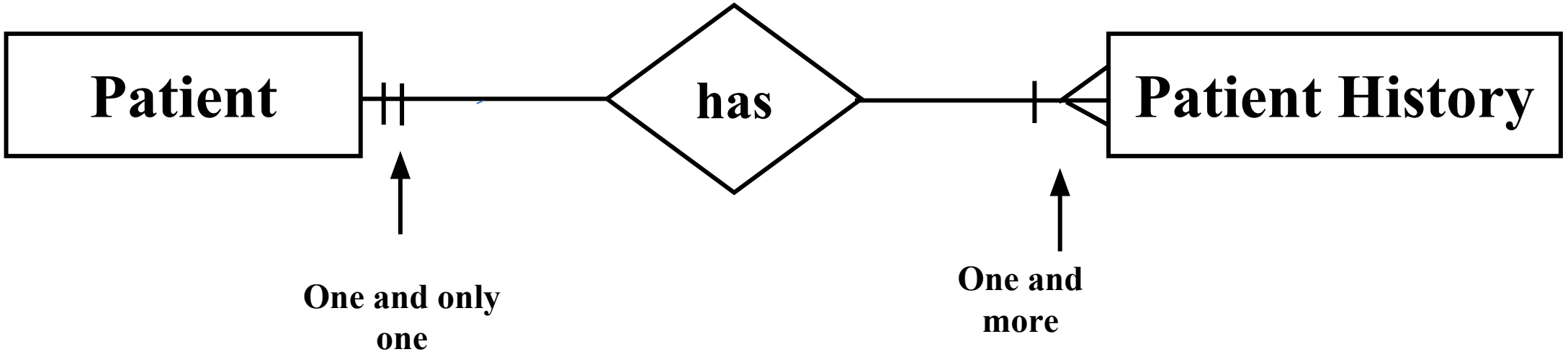
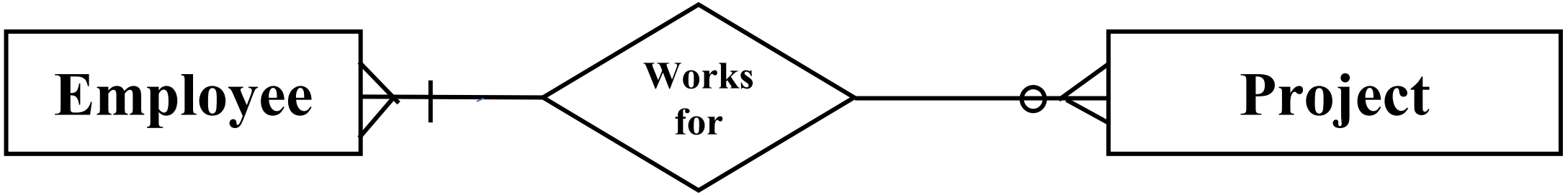


Fig: one to one or one to many relationship

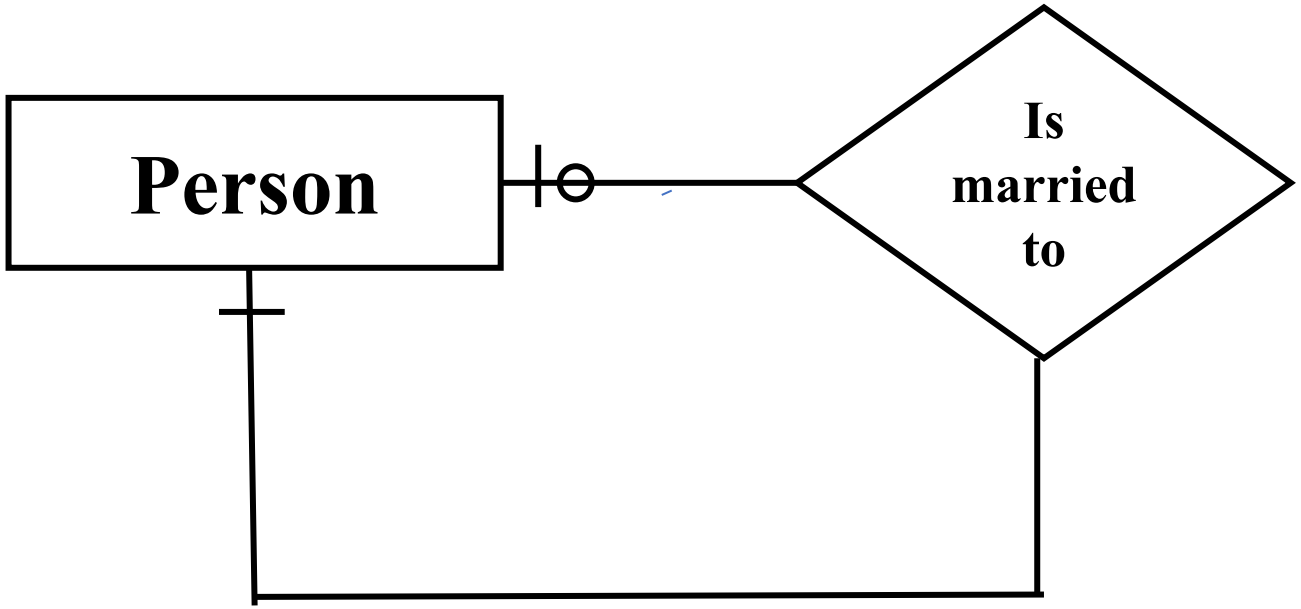
II. One optional one mandatory cardinalities

- One employee works for zero or more projects. Or
- One project is completed by one or more employees.

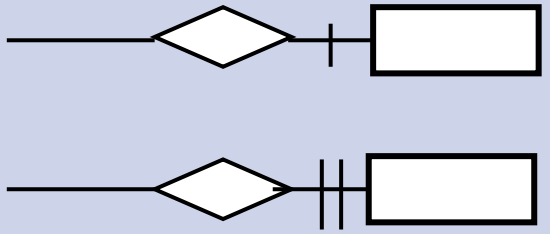
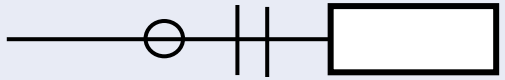
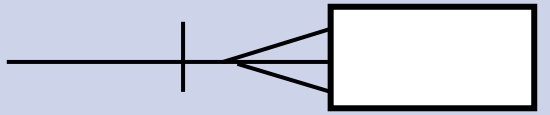
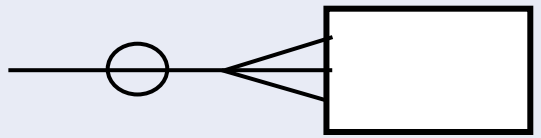
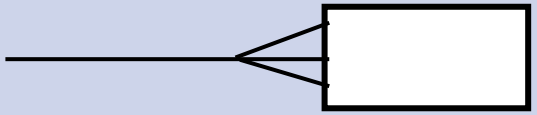


III. Optional cardinalities

- One person is married to one or no person.



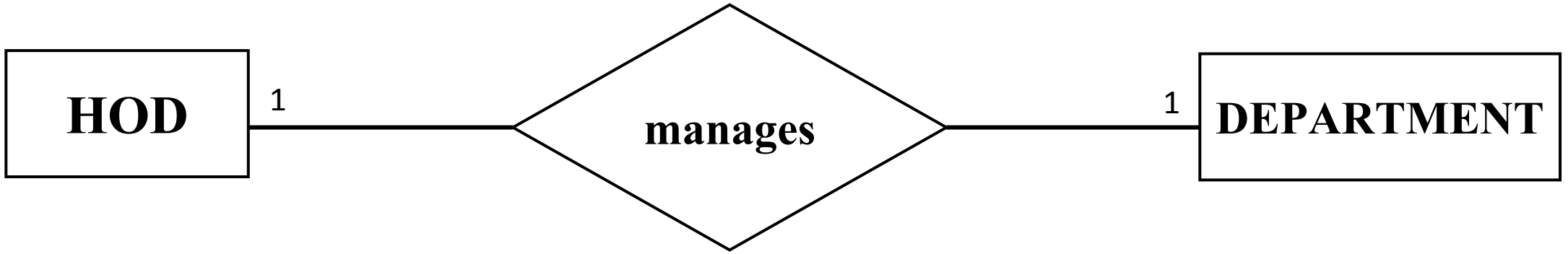
Cardinalities Notation

Cardinality	Minimum Instances	Maximum Instances	Notation
1.Exactly One(Mandatory One/One and only One)	1	1	
2.Zero or one(Optional or 1)	0	1	
3.One or more(Mandatory or many)	1	Many (>1)	
4.Zero or many	0	Many (>1)	
5.More than one	>1	>1	

- Cardinalities can be used as numbers also.

Relationship can be studied under following headings:

1.One to one (1:1)



- One HOD manages one department.

HOD

Hod_id	Hod_name	Hod_Address
1	ram	gongabu
2	sita	putalilsadak
3	shyam	kalanki

Manages

Hod_id	Dept_id
1	23
2	24
3	35

Department

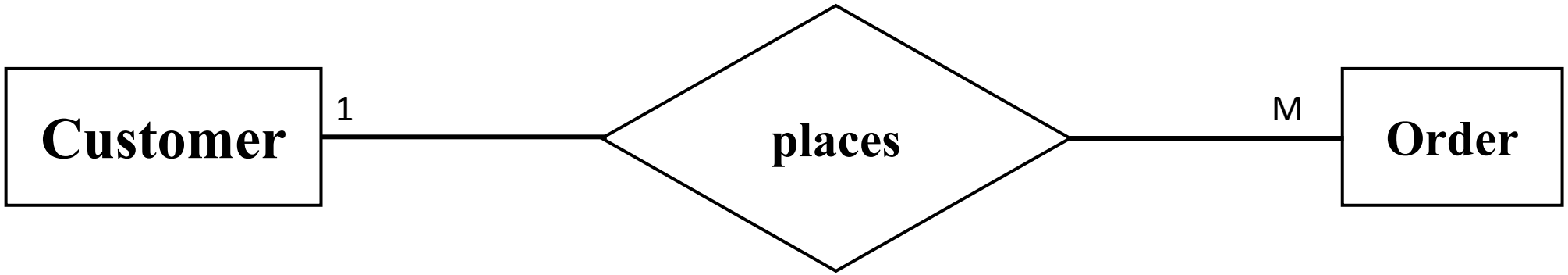
Dept_id	Dpt_name	Dpt_location
23	Computer	A507
24	Archi	B304
35	Civil	C111

- Here , manages table can have PK either Hod_id or Dept_id .And tables can be reduced as below:

Hod_id	Dept_id	Hod_name	Hod_Address
1	23	ram	gongabu
2	24	sita	putalilsadak
3	35	shyam	kalanki

Dept_id	Hod_id	Dpt_name	Dpt_location
23	1	Computer	A507
24	2	Archi	B304
35	3	Civil	C111

2. One to many (1:M)



- One Customer places many order

Customer

Cust_id	Cust_name	Cust_Address
1	ram	gongabu
2	sita	putalilsadak
3	shyam	kalanki

Places

Cust_id	ordert_id
1	23
1	24
2	35

Order

Order_id	quantity	Item_name
23	1	watch
24	2	shoe
35	3	jacket

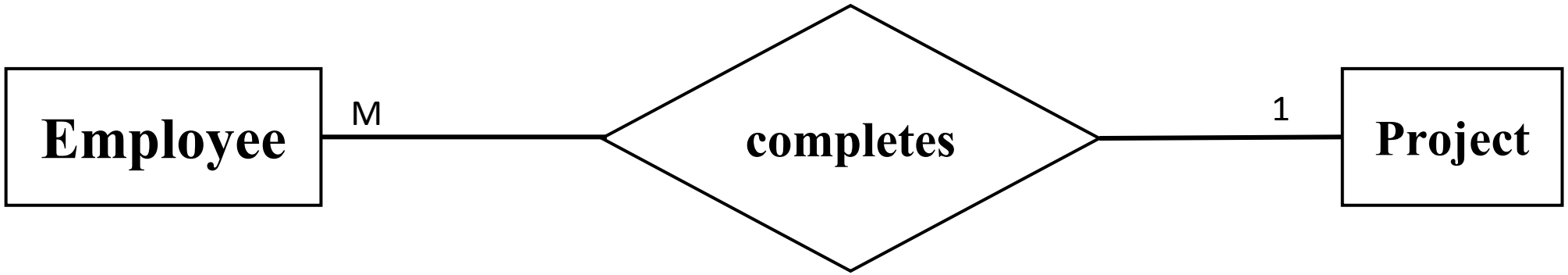
FK **FK**

- Here , places table can have Order_id as PK . And tables can be reduced as below:

Cust_id	Cust_name	Cust_Address
1	ram	gongabu
2	sita	putalilsadak
3	shyam	kalanki

Order_id	Cust_id	quantity	Item_name
23	1	Computer	A507
24	2	Archi	B304
35	3	Civil	C111

3. Many to one (M:1)



- Many employees completes one project.

Employee

Emp_id	Emp_name	Emp_Address
1	ram	gongabu
2	sita	putalilsadak
3	shyam	kalanki

Completes

Emp_id	project_id
1	23
2	23
3	24

FK FK

Project

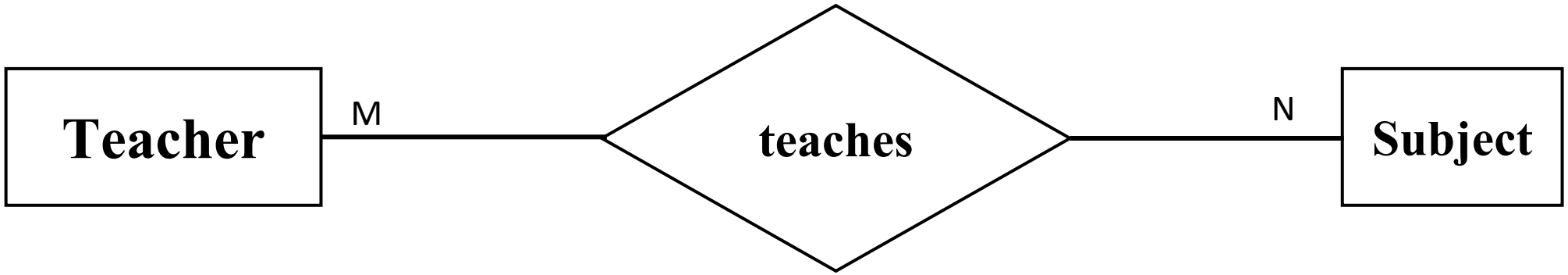
Project_id	Location	Cost
23	pqr	1 crore
24	Xyz	50 lakhs
35	mno	32 lakhs

- Here , completes table can have Emp_id as PK . And tables can be reduced as below:

Emp_id	Project_id	Emp_name	Emp_Address
1	23	ram	gongabu
2	23	sita	putalilsadak
3	24	shyam	kalanki

Project_id	Location	Cost
23	pqr	1 crore
24	Xyz	50 lakhs
35	mno	32 lakhs

4. many to many (M:N)



- Many Teacher teaches many Student.

Teachers

T_id	T_name	T_Address
1	ram	gongabu
2	sita	putalilsadak
3	shyam	kalanki

teaches

T_id	Sub_id
1	BEG123Co
2	BEG13Arc
3	BEG101Ci

FK **FK**

Subject

Sub_id	Sub_name	S_creditHrs
BEG123Co	pqr	45
BEG13Arc	Xyz	50
BEG101Ci	mno	55

- Here , teaches table has T_id and Sub_id as PK(composite key) . And tables cannot be reduced at all .In this case the relationship table is said to be **Associative Entity.**

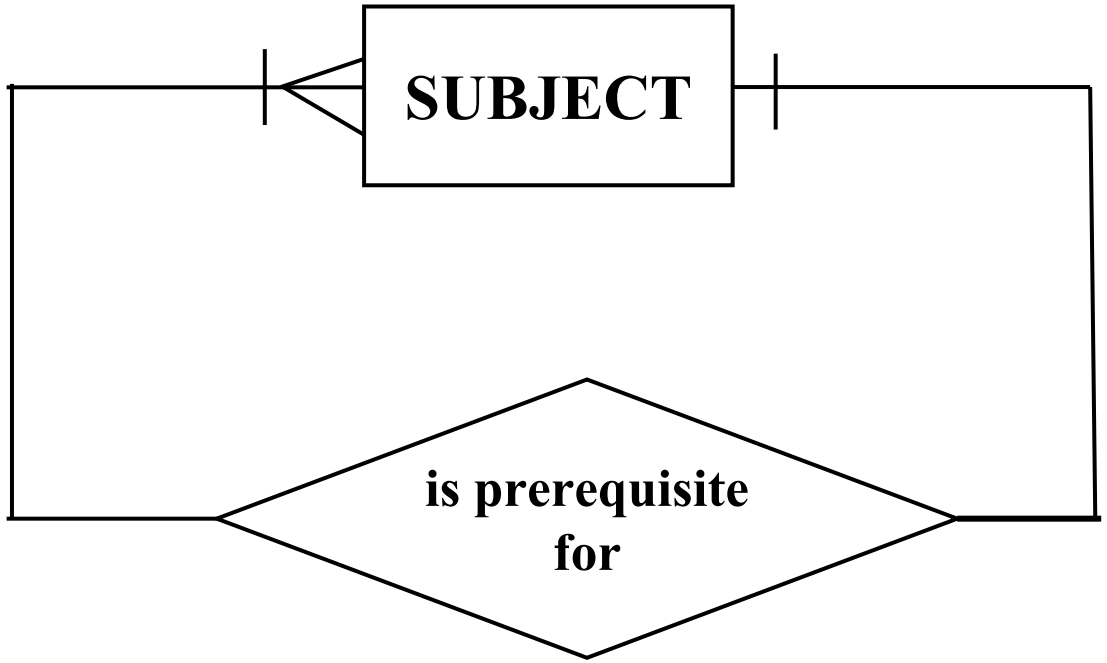
Degree of relationship

- The number of entities that participate in relationship is called the degree of the relationship.
- Types of relationship:
  - Relationship can be of following kinds in terms of their degrees:
    1. Unary relationship(degree-1)
    2. Binary relationship(degree-2)
    3. Ternary relationship(degree-3)
    4. High degree or n-ary relationship(degree-4 or above)

1.Unary relationship

- Relationship between the instances of same(**ONE**) entity.

  Example: one subject is prerequisite for one or more subjects.



2.Binary relationship

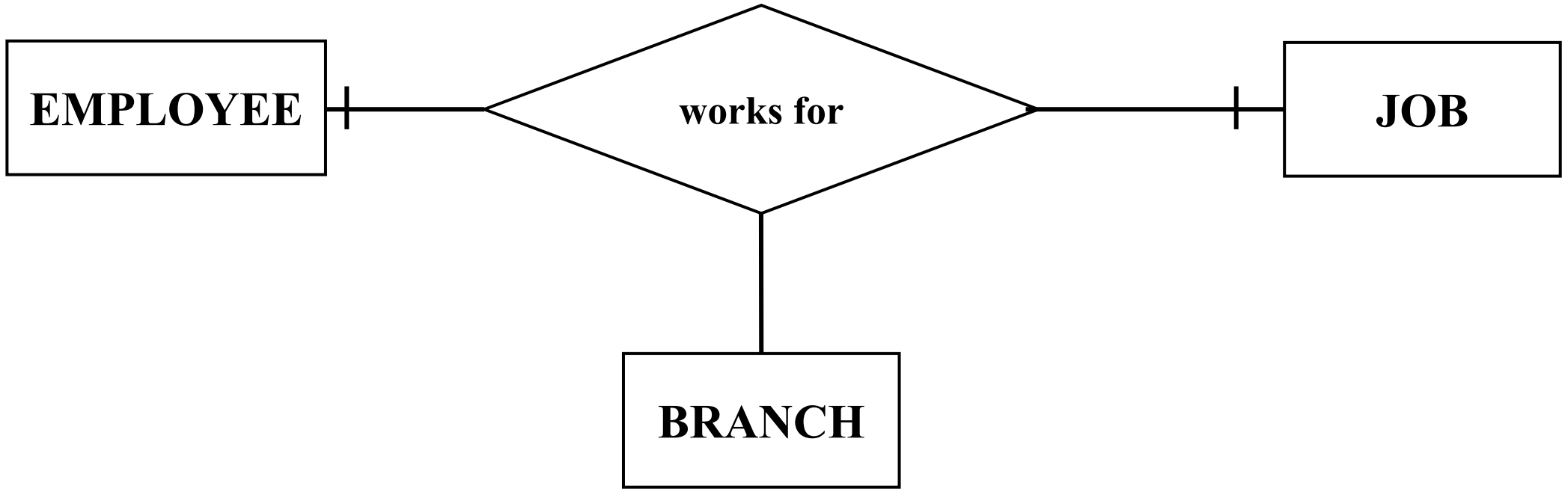
- Relationship between the instances of **TWO** entities.

  Example: one student register for one or many courses or one course can be registered by none or many students.

3.Ternary relationship

- Relationship between the instances of **THREE** entities.

  Example: one employee works for one job at branch



Steps to draw ER-Diagram

1. Identify all the entities.
2. Identify all the relationship between the entities using relationship matrix .
3. Draw a rough ER-diagram using identified entities and relationship.
4. Fill the cardinality number among them.
5. Find unique identifier (Primary key and foreign key in case of related entities)
6. Eliminate many to many relationship .
    - Many to many relationship conceals a hidden entity called as associative entity giving rise to one to many relationship with the existing entity.

Example of ERD

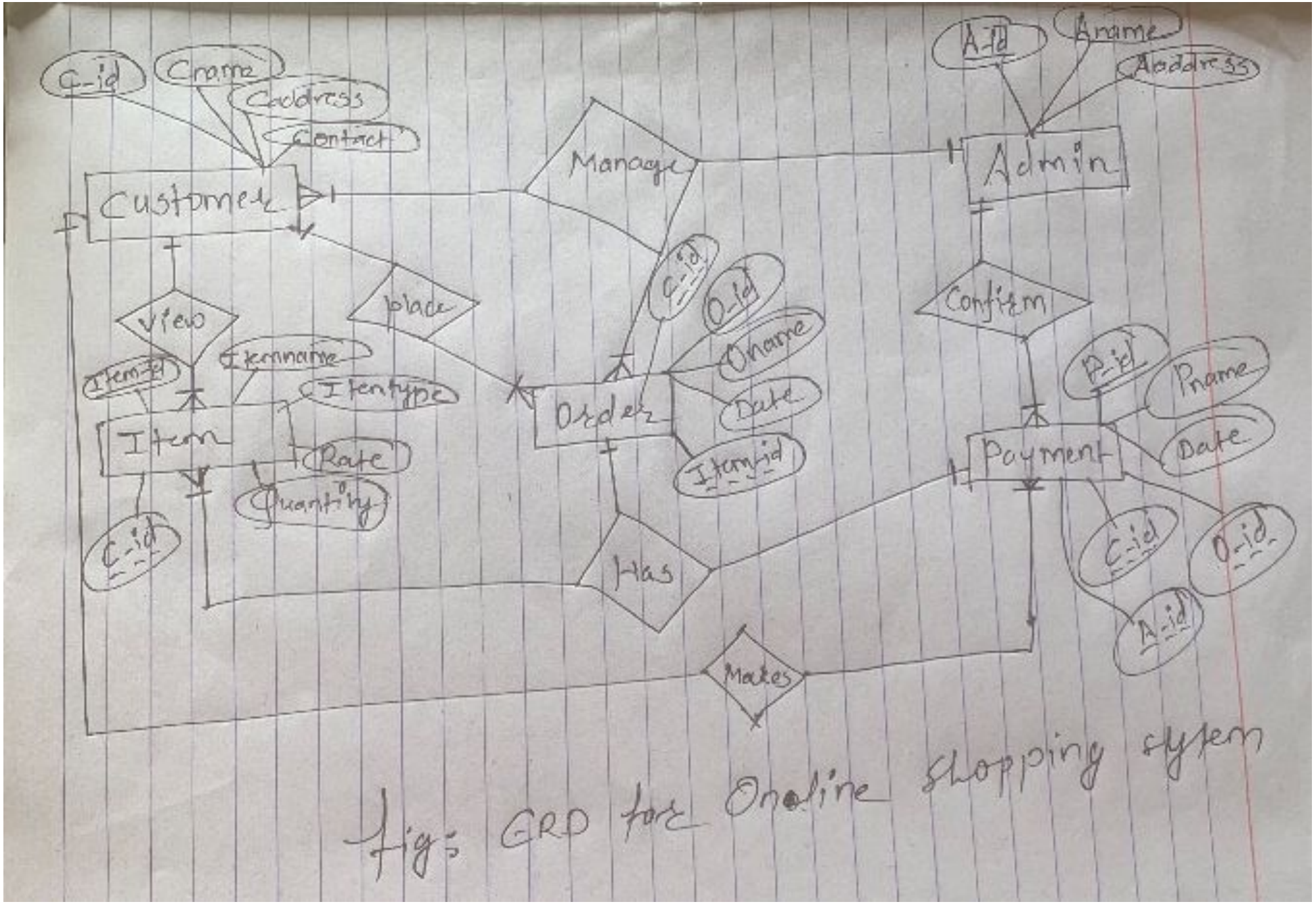


Fig: ERD for Online Shopping system

Example of ERD

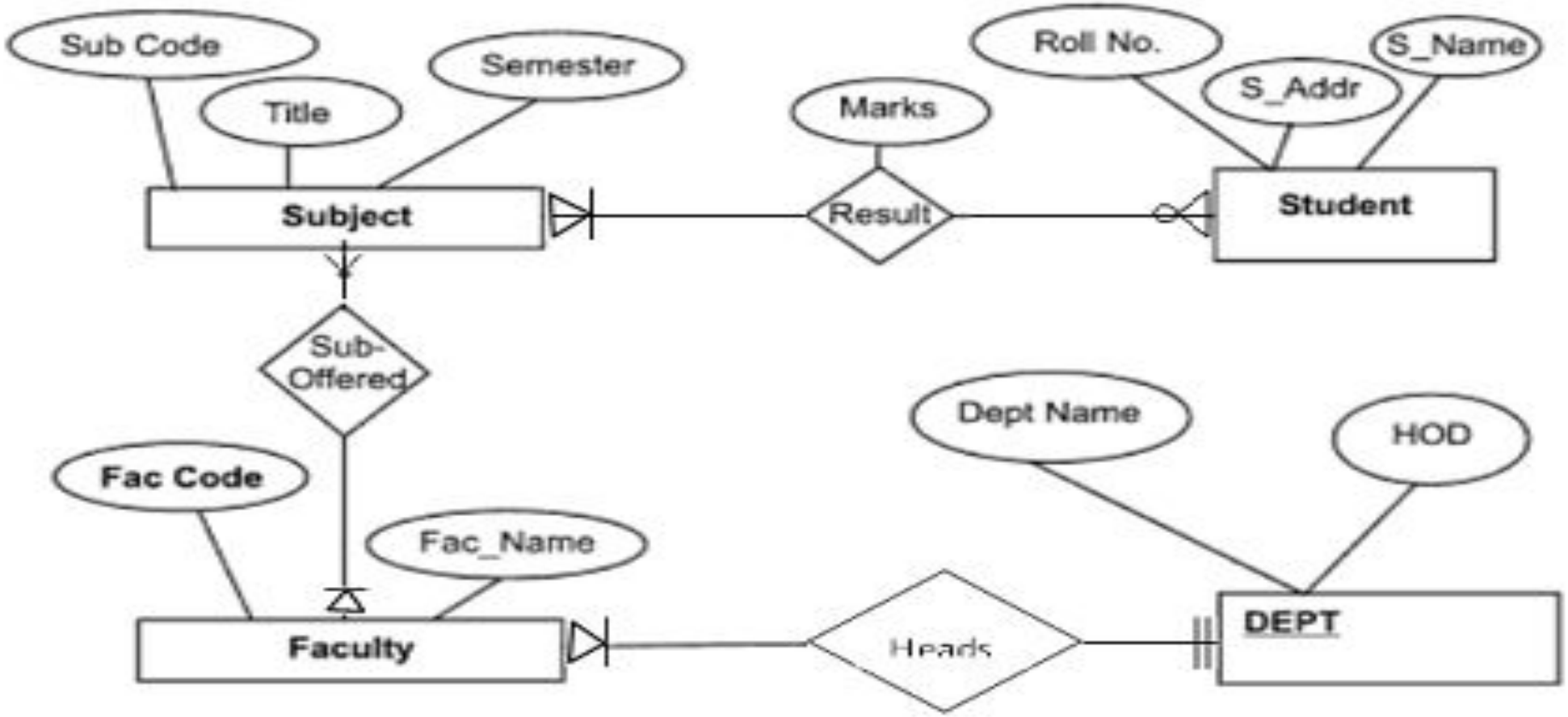
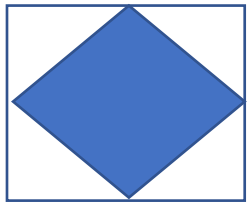


Figure: ERD for College management system

Associativity Entity

- Associative entity is a type of entity that associate with the instance of one or more entity type with one another.
- Presence of one or more attributes on the relationship suggest the designer that the relationship should be represented as its type.
- Concepts:
  - Associative entity represents an association between two fundamental entities that has a potential many-to-many cardinality or that holds some attributes.
  - It can be concrete, such as the birth place, or it can be more generic, representing commonalities between a number of concrete associative entities, such as the root parents.
- Associative entity is used to implement M:N relationship between two or more entities.
- This entity inherits its primary key from the entities to which it is associated(PARENTS Entities).
- Also known as **Composite** or **Bridge** entities.

Symbol: 

Example-1

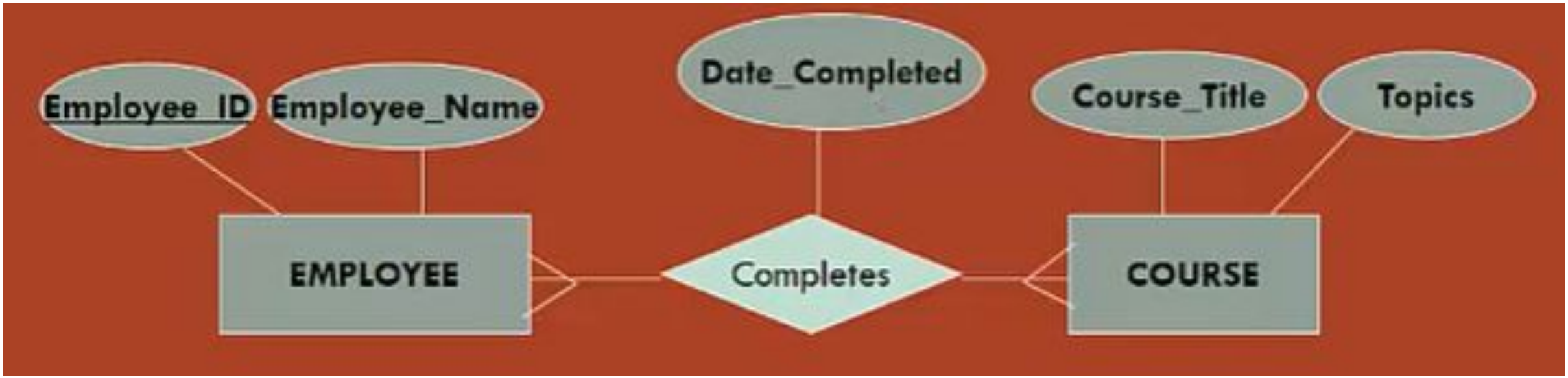
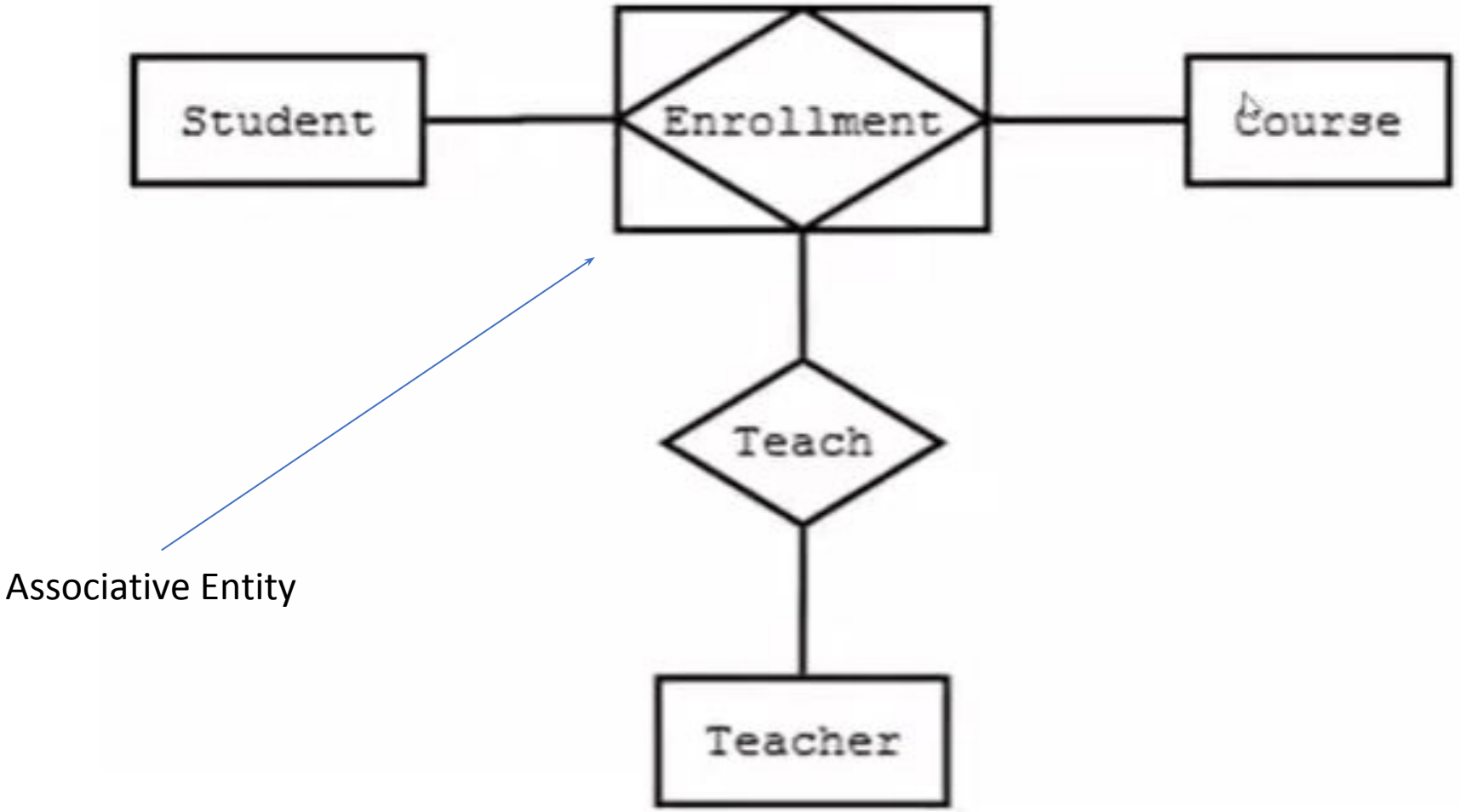

Employee_ID
Employee_Name
Date_Completed
Course_Title
Topics
EMPLOYEE
Completes
COURSE

Fig: ERD with associative entity

- Here, CERTIFICATE is associative entity.
- EMPLOYEE will get certificate if he/she complete the course within the date.

Example-2


Student
Enrollment
Course
Teach
Teacher
Associative Entity

Extended ER-features

1. **Generalization:**
   - Generalization is the process of extracting common properties from a set of entities and create a generalized entity from it.
   - It is a bottom-up approach in which two or more entities can be generalized to a higher level entity if they have some attributes in common.

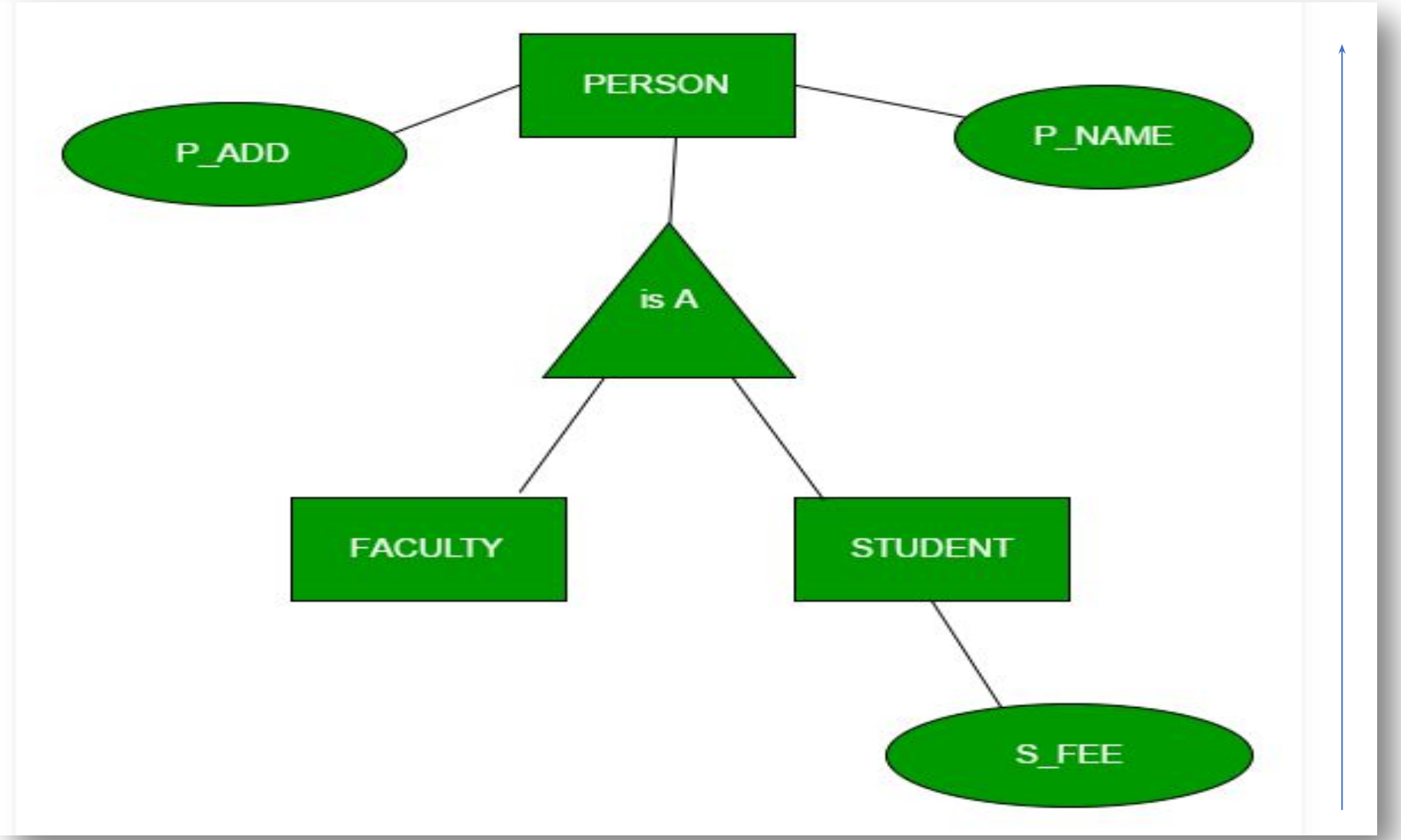


Figure: Generalization example

2. Specialization:

- In specialization, an entity is divided into sub-entities based on their characteristics.
- It is a top-down approach where higher level entity is specialized into two or more lower level entities.

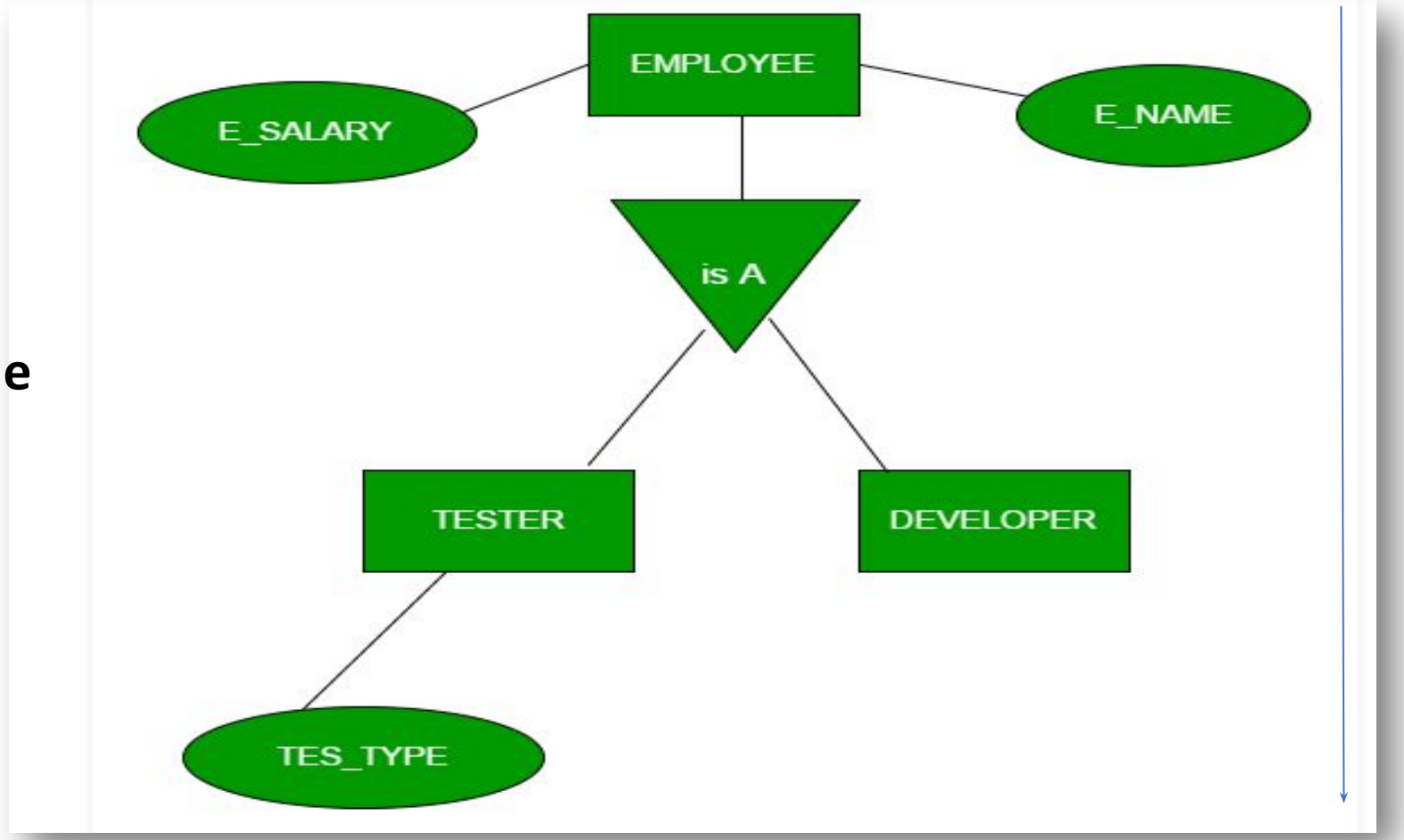
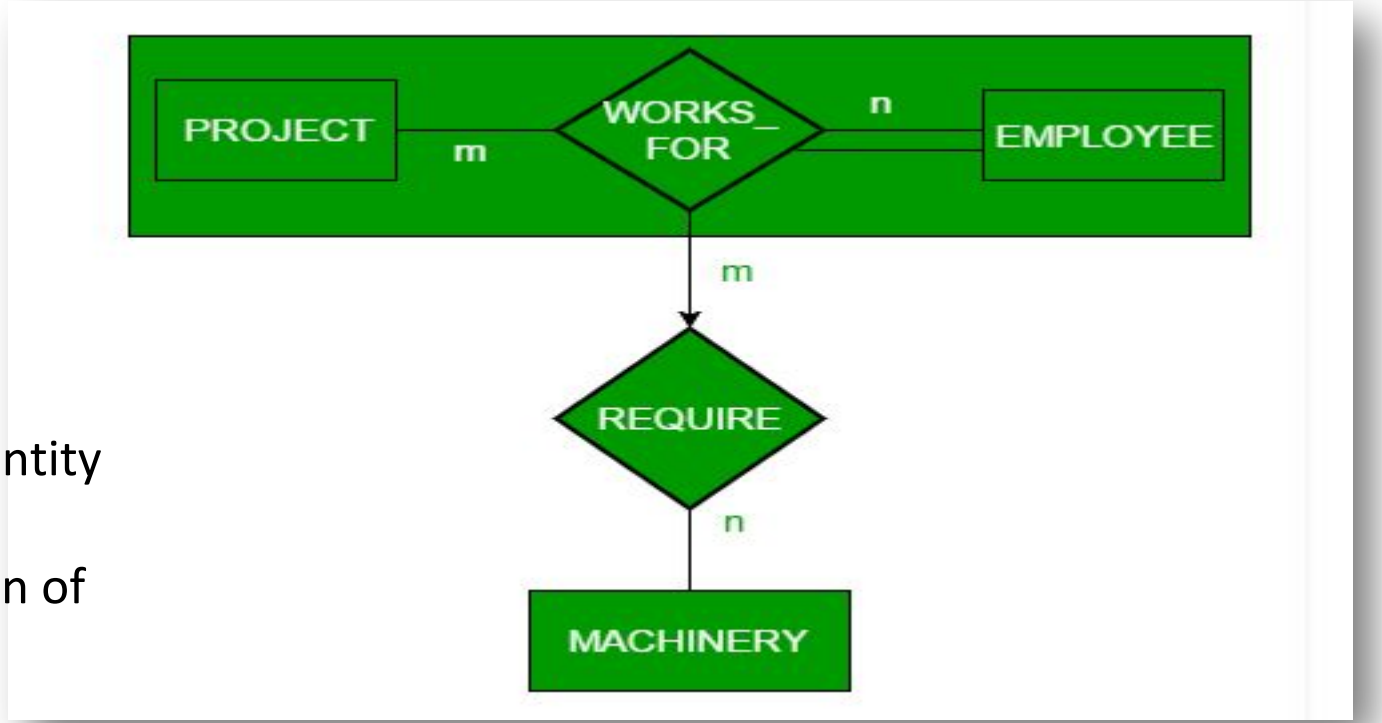


Figure: Specialization example

3. Aggregation:

- An ER diagram is not capable of representing 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.



Lines: link the attributes to entity sets and entity sets to relationship sets.
Doubles lines: indicate the total participation of an entity in a relationship set.

Figure: Aggregation example

Roles of ER-diagram

- It allows us to sketch DB schema designs.
  - It includes some constraints.
- It helps us to understand the system conceptually.
- It is an easy to use graphical tool for modeling data
- It is a GUI representation of the logical structure of a Database
- It helps you to identifies the entities which exist in a system and the relationships between those entities.
- It also helps to design relational DB.

Assignment:

- Explain identifying relationship with example.
- Compare strong entity and weak entity
- Draw ERD for:
  - Library management system
  - Bank management system

Role of CASE in Conceptual Data Modeling