3.3 Structuring System Data Requirements

Introduction; Conceptual Data Modeling (Process, Deliverables and Outcomes); Gathering Information for Conceptual Data Modeling; Introduction to E-R Modeling (Entities, Attributes, Keys and Identifiers, Relationships: Degree, Cardinality, Naming

System Data Requirements

- Introduction: In previous chapter , we learned how to model and analyze data. we learned how to show data stores, or data at rest, in a data flow diagram (DFD). DFDs, use cases, and various processing logic techniques show how, where, and when data are used or changed in an information system, but these techniques do not show the definition, structure, and relationships within the data. Data modeling develops these missing, and crucial, descriptive pieces of a system. The most common format used for data modeling is entity-relationship (E-R) diagramming. A similar format used with object-oriented analysis and design methods is class diagramming, which is included in a special section at the end of this chapter on the object-oriented development approach to data modeling. Data models that use E-R and class diagram notations explain the characteristics and structure of data independent of how the data may be stored in computer memory. A data model is usually developed iteratively, either from scratch or from a purchased data model for the industry or business area to be supported. Information system (IS) planners use this preliminary data model to develop an enterprise-wide data model with very broad categories of data and little detail.

Conceptual Data Modeling

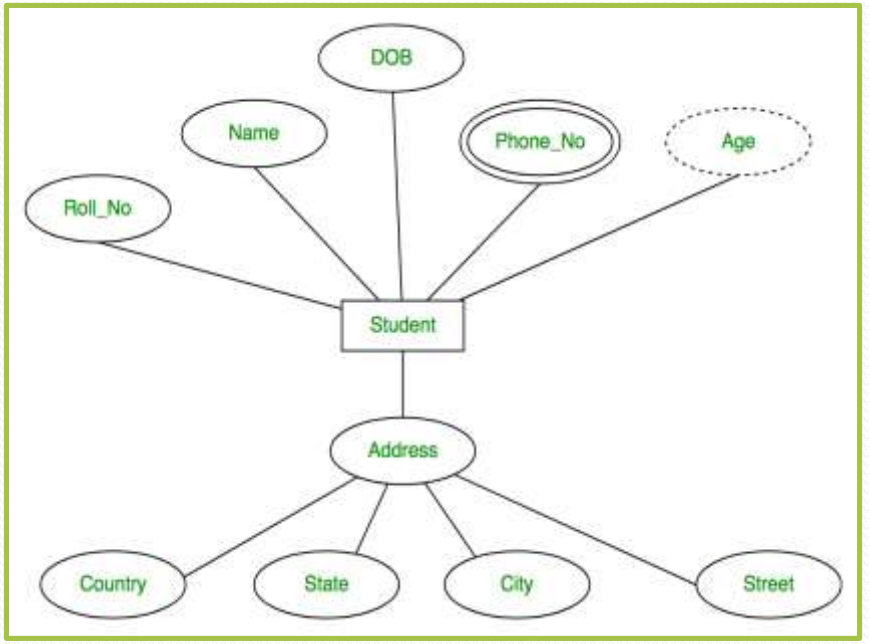
- A conceptual data model is a representation of organizational data. The purpose of a conceptual data model is to show as many rules about the meaning and interrelationships among data as are possible. Conceptual data modeling is typically done in parallel with other requirements analysis and structuring steps during systems analysis .On larger systems development teams, a subset of the project team concentrates on data modeling while other team members focus attention on process or logic modeling. Analysts develop (or use from prior systems development) a conceptual data model for the current system and then build or refine a purchased conceptual data model that supports the scope and requirements for the proposed or enhanced system. The work of all team members is coordinated and shared through the project dictionary or repository. This repository is often maintained by a common Computer- Aided Software Engineering (CASE) or data modeling software tool.

Deliverables and outcomes

- Most organizations today do conceptual data modeling using E-R modeling, which uses a special notation to represent as much meaning about data as possible. Because of the rapidly increasing interest in object-oriented methods, class diagrams using unified modeling language (UML) drawing tools such as IBM's Rational products or Microsoft Visio are also popular.
  - The primary deliverable from the conceptual data modeling step within the analysis phase is an E-R diagram.
  - The other deliverable from conceptual data modeling is a full set of entries about data objects that will be stored in the project dictionary, repository, or data modeling software.The repository is the mechanism to link data, process and logic models of an information system.

Entity Relationship (E-R) Model

- Entity relationship model also called e-r model and it is a high level model.
- This model is used to define the data elements and relationship for a specified system.
- It develops a conceptual design for the database. It also develops a very simple and easy to design view of data
- In ER modeling, the database structure is portrayed as a diagram called an entity-relationship diagram

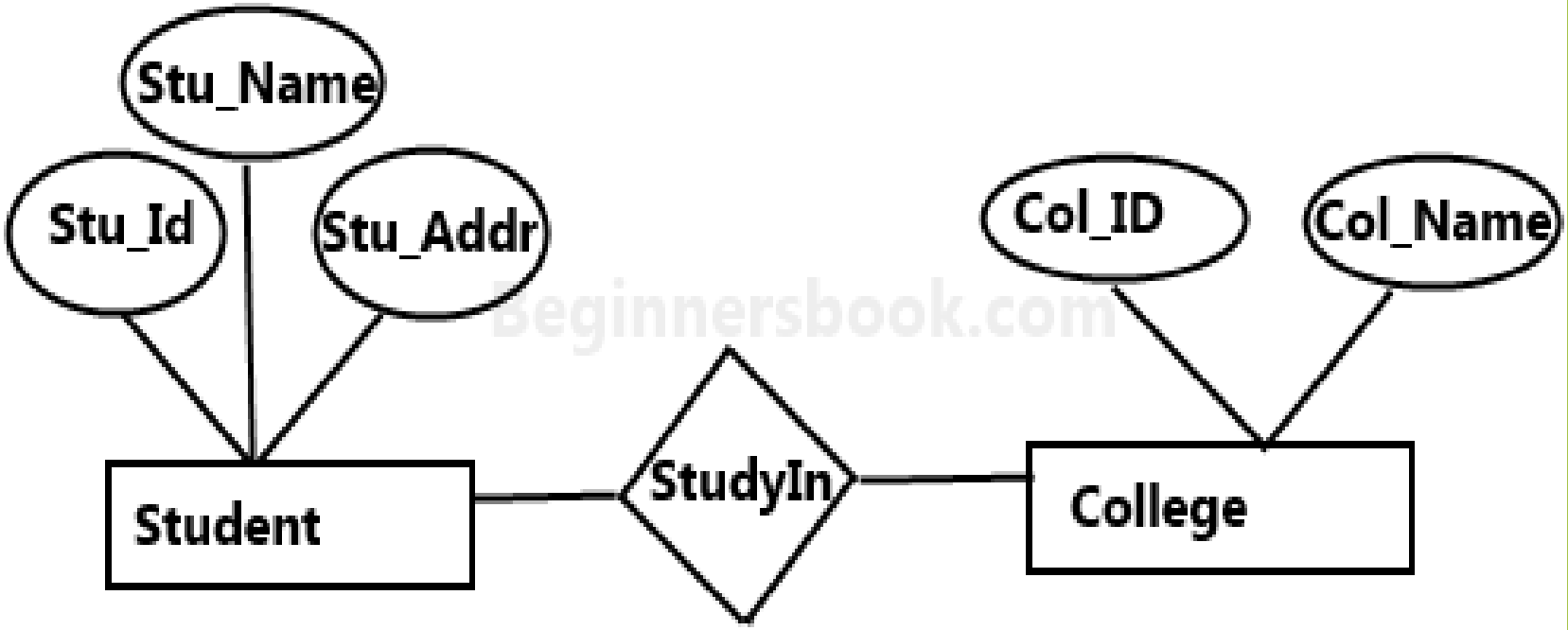


Entity-relationship Diagram (ERD)

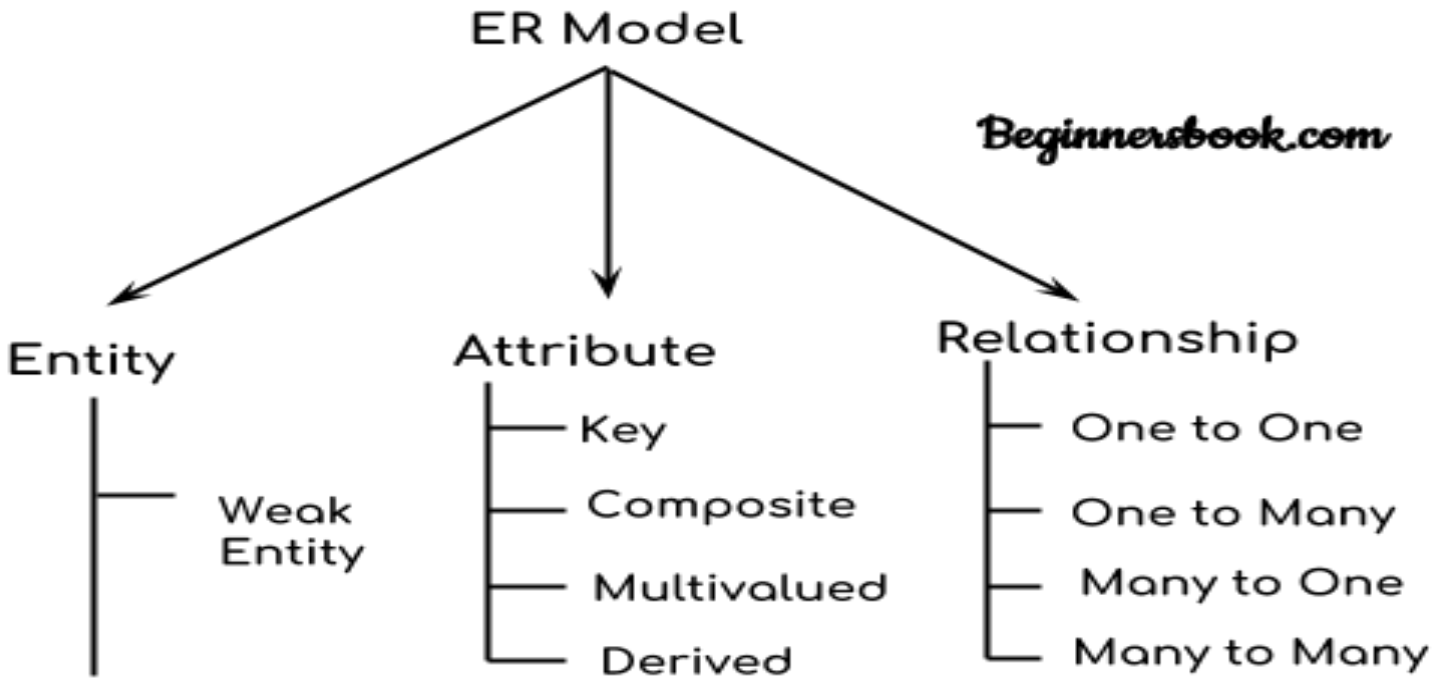
- **ENTITY-RELATIONSHIP DIAGRAM (ERD)** displays the relationships of entity set stored in a database.
- In other words, we can say that ER diagrams help you to explain the logical structure of databases.
- At first look, an ER diagram looks very similar to the flowchart. However, ER Diagram includes many specialized symbols, and its meanings make this model unique.
- The purpose of ER Diagram is to represent the entity framework infrastructure.

Facts about ER Diagram Model:

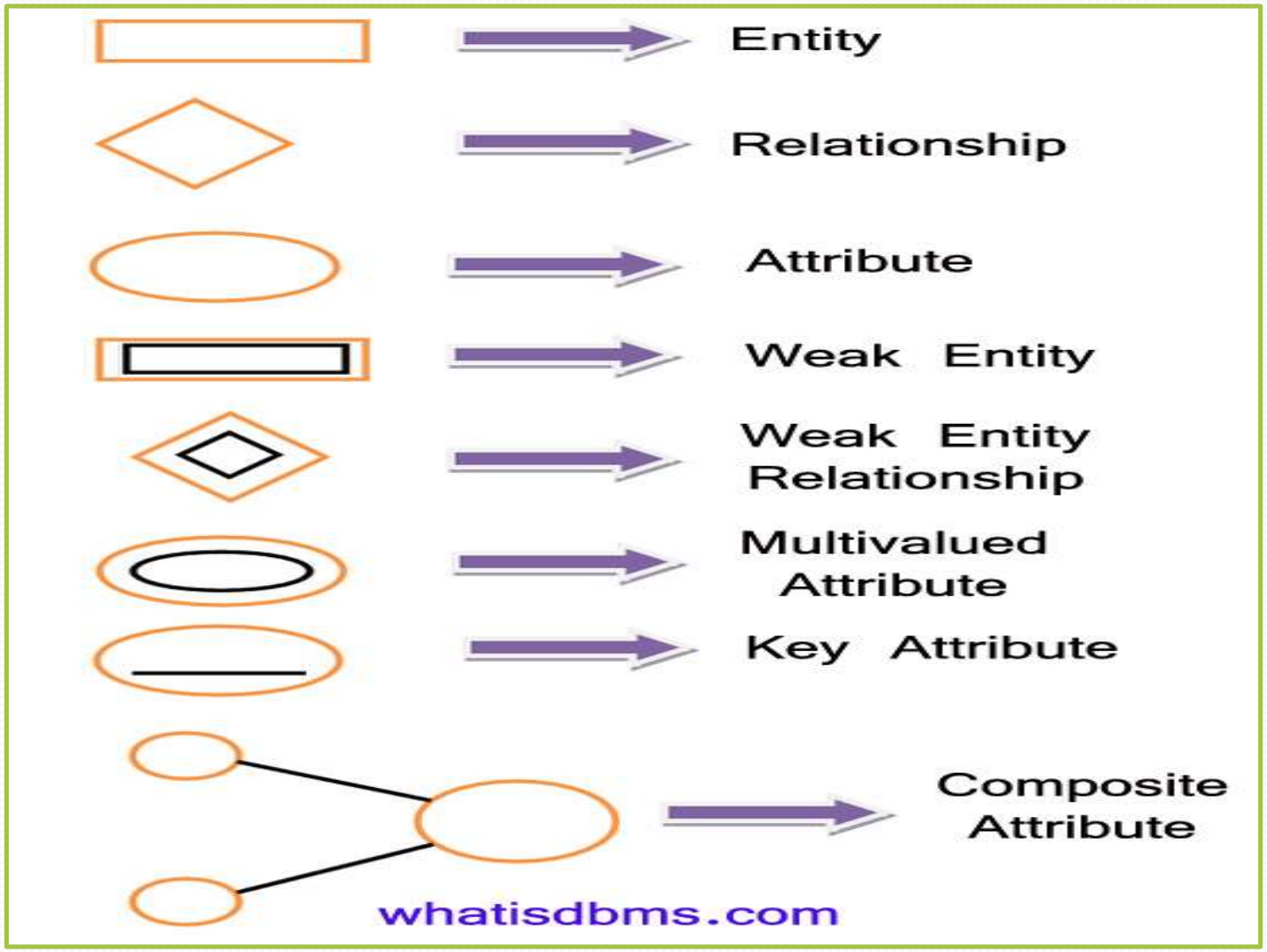
- ER model allows you to draw Database Design
- It is an easy to use graphical tool for modeling data
- Widely used in Database Design
- It is a GUI representation of the logical structure of a Database
- It helps you to identifies the entities which exist in a system and the relationships between those entities



Sample E-R Diagram



Components of ER Diagram


Entity
Relationship
Attribute
Weak Entity
Weak Entity Relationship
Multivalued Attribute
Key Attribute
Composite Attribute
whatisdbms.com

Entity

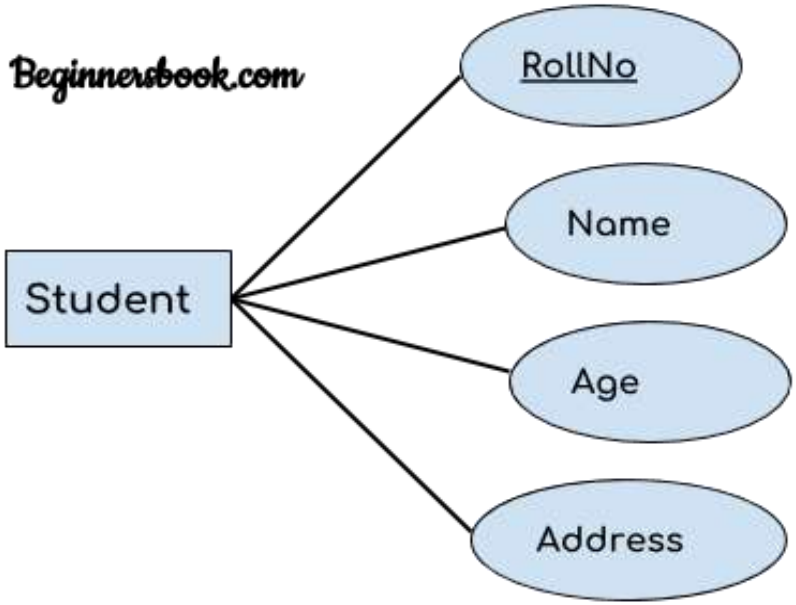
- A real-world thing either living or non-living that is easily recognizable and nonrecognizable.
- It is anything in the enterprise that is to be represented in our database.
- It may be a physical thing or simply a fact about the enterprise or an event that happens in the real world.
- An entity is represented as **rectangle** in an ER diagram.
- **Examples of entities:**
  - **Person:** Employee, Student, Patient
  - **Place:** Store, Building
  - **Object:** Machine, product, and Car
  - **Event:** Sale, Registration, Renewal
  - **Concept:** Account, Course

Weak Entity

- An entity that cannot be uniquely identified by its own attributes and relies on the relationship with other entity is called weak entity.
- The weak entity is represented by a **double rectangle**. For example – a bank account cannot be uniquely identified without knowing the bank to which the account belongs, so bank account is a weak entity.

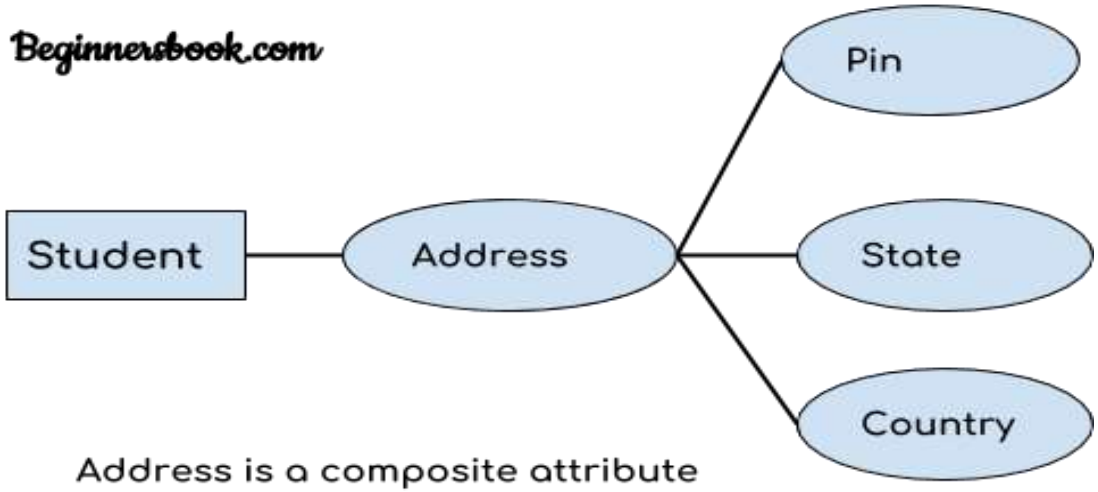
Key attribute

- A key attribute can uniquely identify an entity from an entity set. For example, student roll number can uniquely identify a student from a set of students. Key attribute is represented by oval same as other attributes however the **text of key attribute is underlined**.



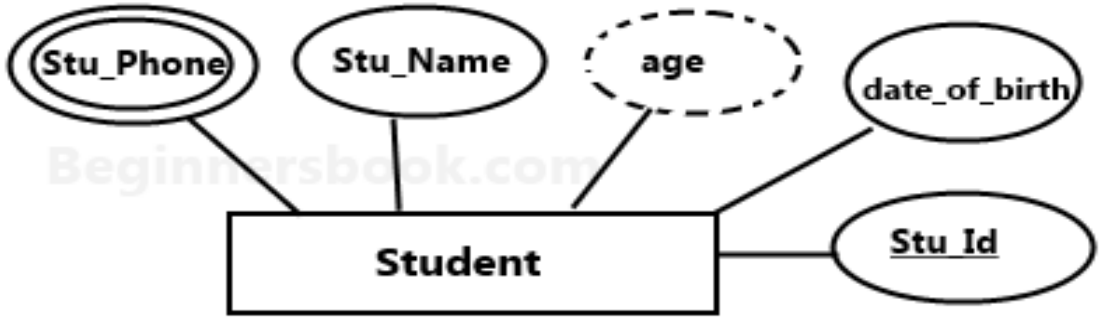
Composite attribute

- An attribute that is a combination of other attributes is known as composite attribute. For example, In student entity, the student address is a composite attribute as an address is composed of other attributes such as pin code, state, country.



Address is a composite attribute

- **Multivalued attribute:** An attribute that can hold multiple values is known as multivalued attribute. It is represented with **double ovals** in an ER Diagram. For example – A person can have more than one phone numbers so the phone number attribute is multivalued.
- **Derived attribute:** A derived attribute is one whose value is dynamic and derived from another attribute. It is represented by **dashed oval** in an ER Diagram. For example – Person age is a derived attribute as it changes over time and can be derived from another attribute (Date of birth).



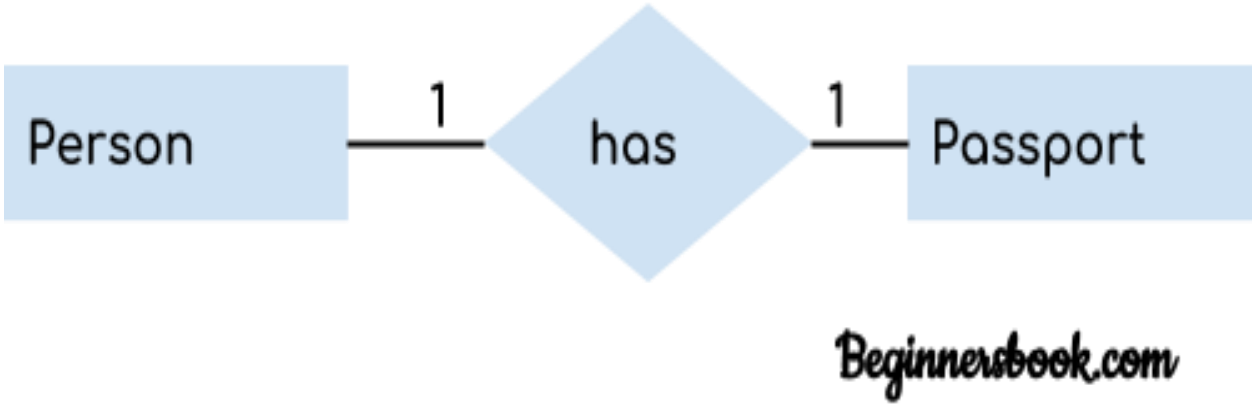
- **Relationship**
  - A relationship is represented by diamond shape in ER diagram
  - It shows the relationship among entities.

One to Many Relationship

- **Cardinality**
  - Defines the numerical attributes of the relationship between two entities or entity sets.
  - In data modeling terms, cardinality is how one table relates to another.
  - There are four types of relationships:
    - One to One
    - One to Many
    - Many to One
    - Many to Many

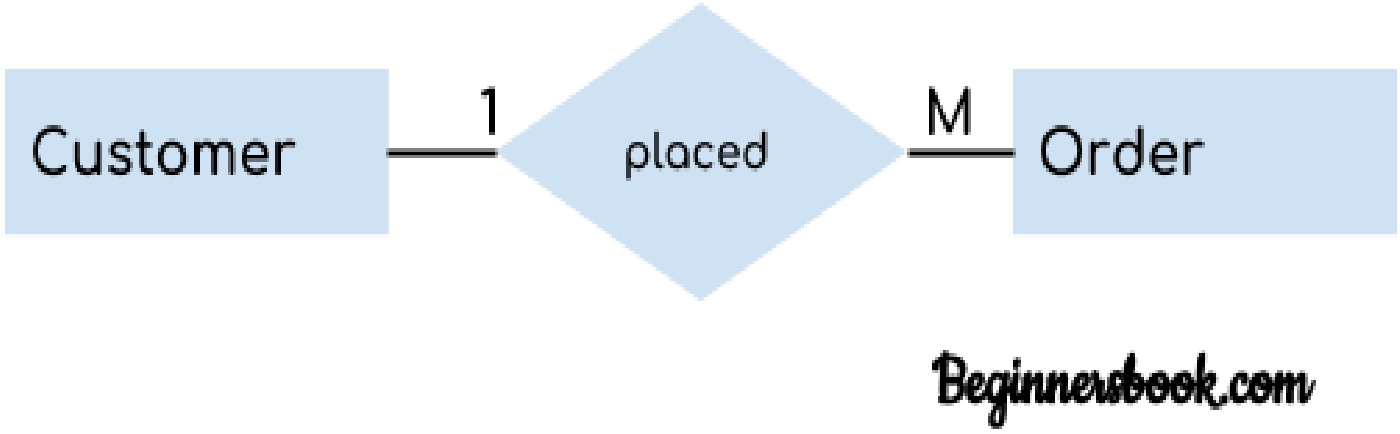
One to One Relationship

- When a single instance of an entity is associated with a single instance of another entity then it is called one to one relationship.
- For example, a person has only one passport and a passport is given to one person.



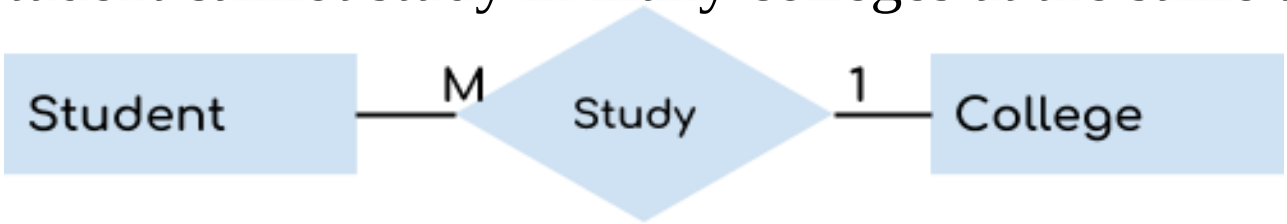
One to Many Relationship

- When a single instance of an entity is associated with more than one instances of another entity then it is called one to many relationship.
- For example – a customer can place many orders but a order cannot be placed by many customers.



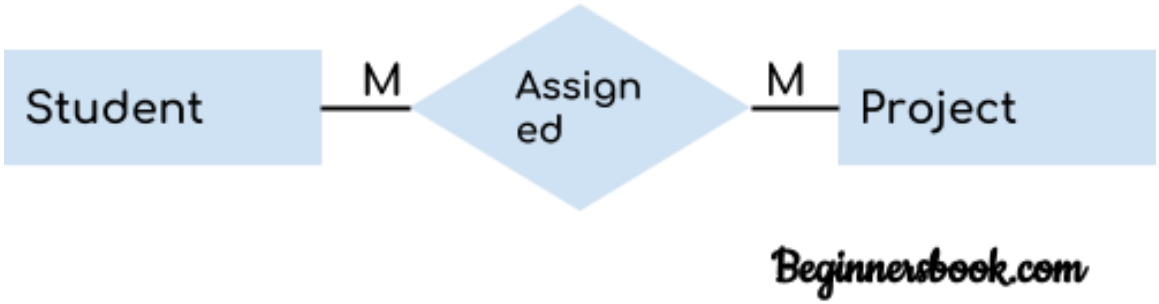
Many to One Relationship

- When more than one instances of an entity is associated with a single instance of another entity then it is called many to one relationship.
- For example – many students can study in a single college but a student cannot study in many colleges at the same time.



Many to Many Relationship

- When more than one instances of an entity is associated with more than one instances of another entity then it is called many to many relationship.
- For example, a can be assigned to many projects and a project can be assigned to many students.



Cardinality Relationship

C Computer
S Social
E Education

Cardinality – The number of entities to which another entity can be associated through a relationship

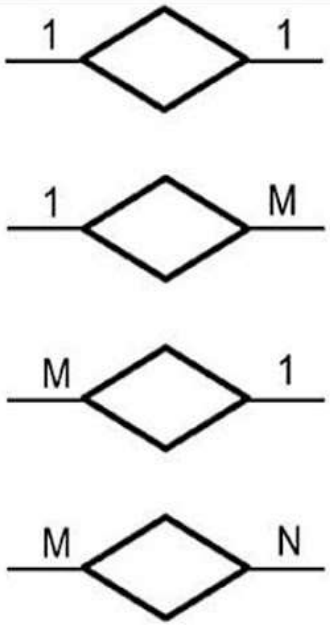
The diagrams on the right show, in order:

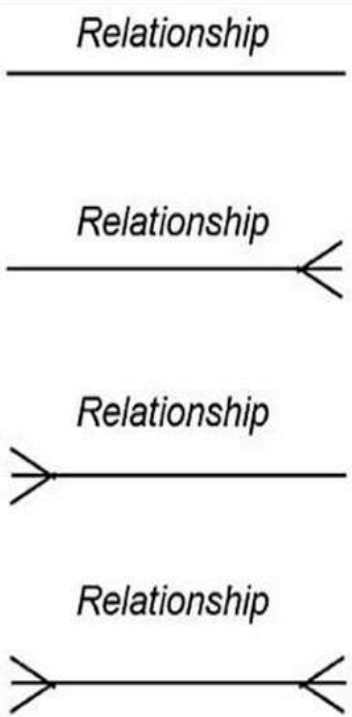
one-to-one

one-to-many

many-to-one

many-to-many





Entity Type

- The entity type is a collection of the entity having similar attributes.
- In the below student table example, we have each row as an entity and they are having common attributes i.e each row has its own value for attributes Roll_no, Age, Student_name and Mobile_no.

Student

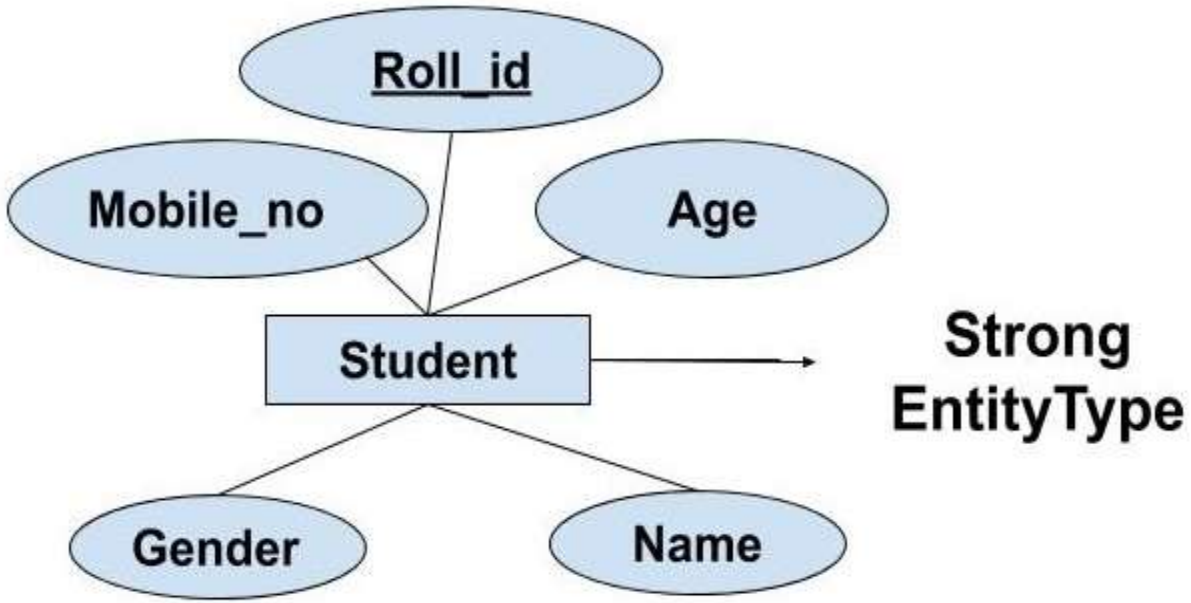
Roll_no	Student_name	Age	Mobile_no
1	Andrew	18	7089117222
2	Angel	19	8709054568
3	Priya	20	9864257315
4	Analisa	21	9847852156

Entity

- **Types of Entity type**
  - Strong Entity Type
  - Weak Entity Type

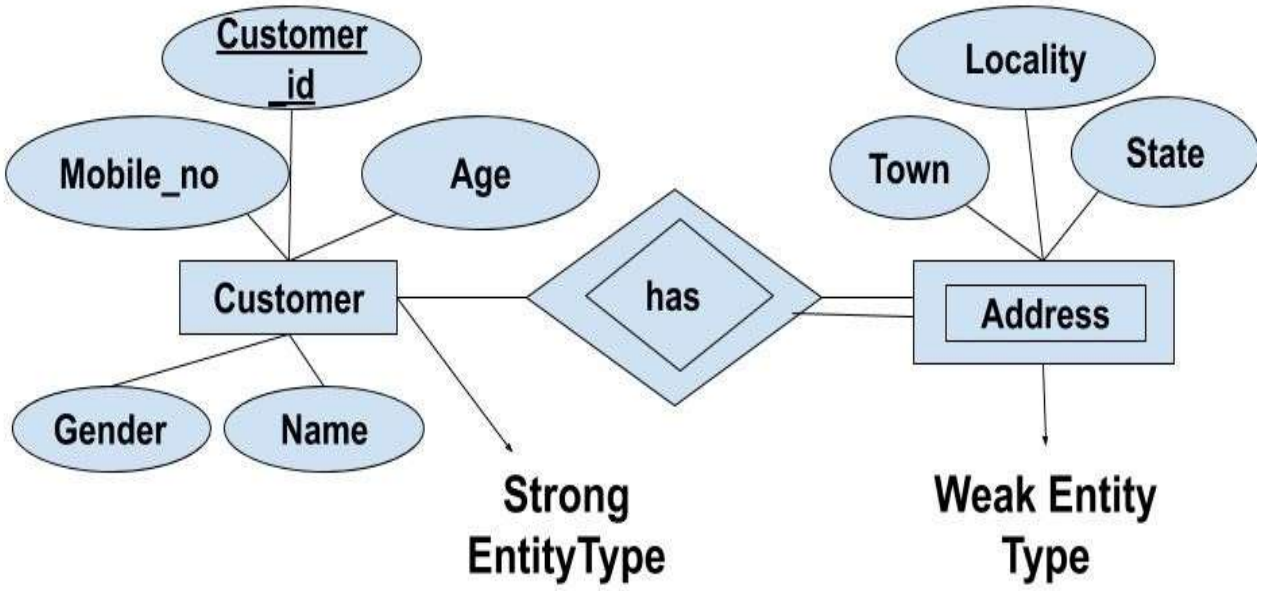
Strong Entity Type: Strong entity are those entity types which has a key attribute.

- The primary key helps in identifying each entity uniquely. It is represented by a rectangle.



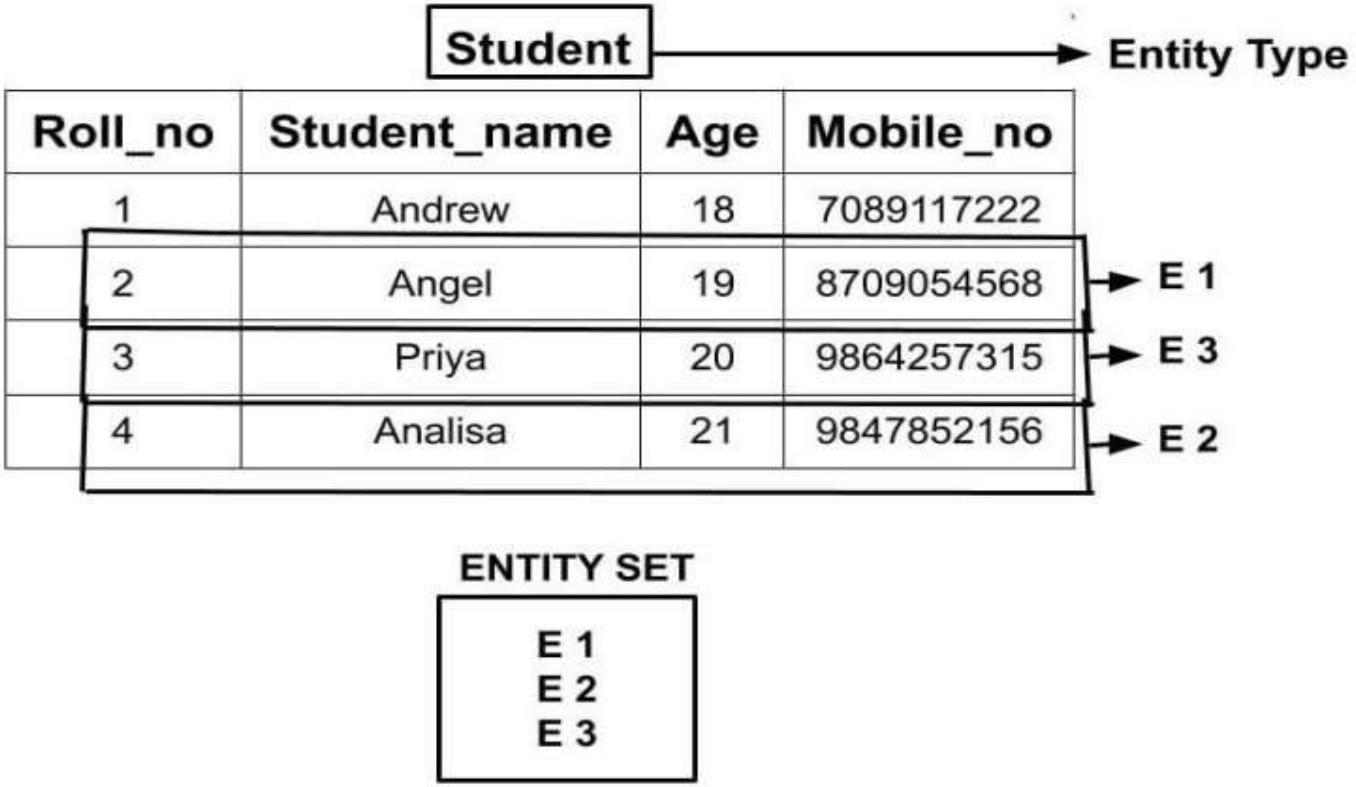
Weak Entity Type: Weak entity type doesn't have a key attribute.

- Weak entity type can't be identified on its own.
- It depends upon some other strong entity for its distinct identity.
- This can be understood with a real-life example. There can be children only if the parent exits.
- A weak entity is represented by a double outlined rectangle. The relationship between a weak entity type and strong entity type is called an identifying relationship and shown with a double outlined diamond instead of a single outlined diamond.

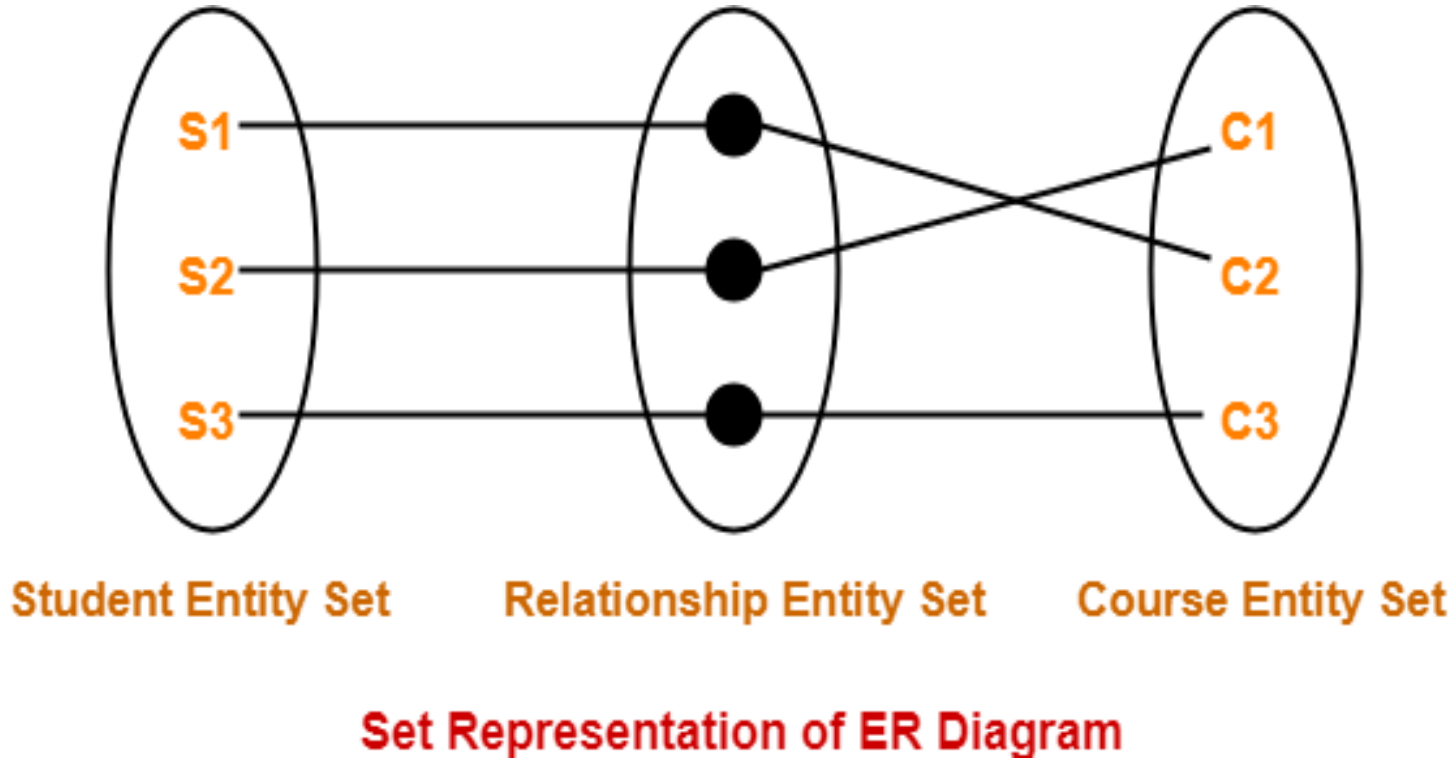


Entity Set

- Entity Set is a collection of entities of the same entity type.



- **Relationship set** is a set of relationship of same type
- Example



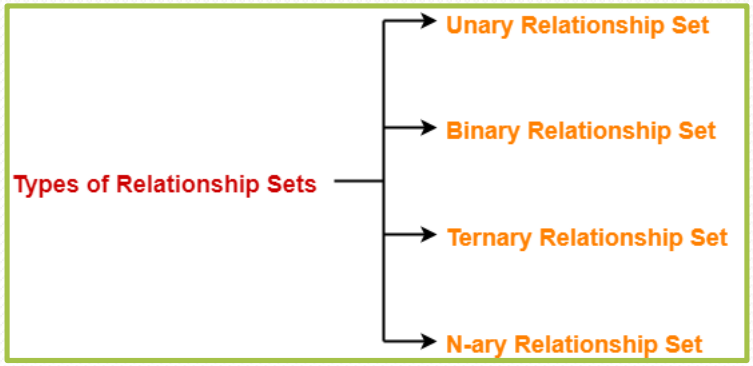
Set Representation of ER Diagram

Degree of relationship set

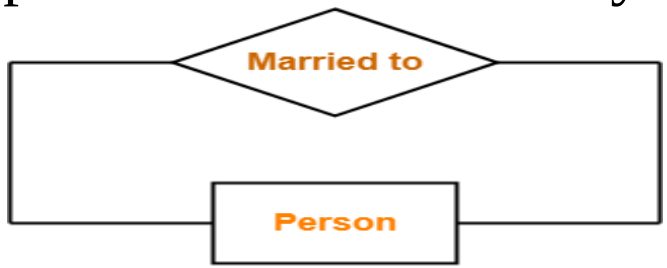
- The number of entity sets that participate in a relationship set is termed as the degree of that relationship set.

Degree of a relationship set = Number of entity sets participating in a relationship set

Types of Relationship set



- **Unary relationship set** is a relationship set where only one entity set participates in a relationship set.
- Example : One person is married to only one person

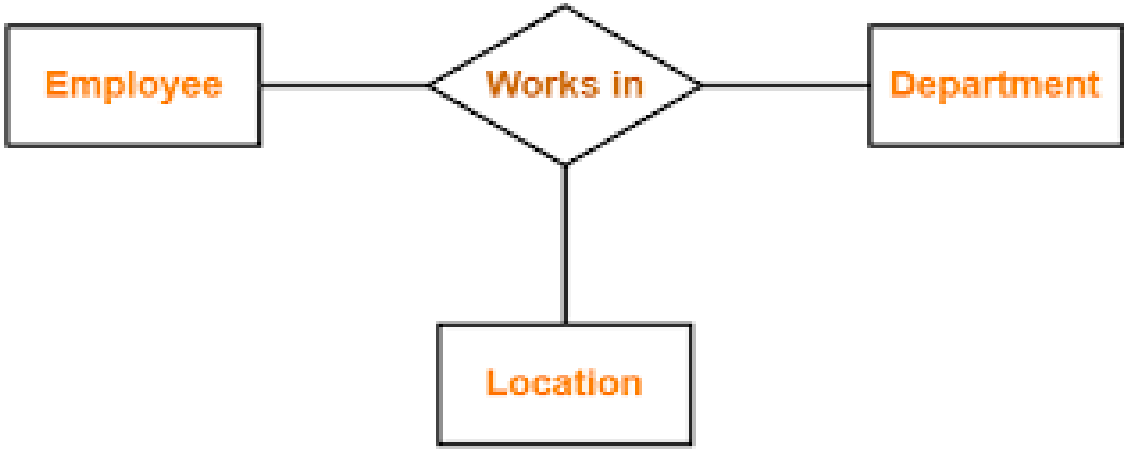


Unary Relationship Set

- **Binary relationship set** is a relationship set where two entity sets participate in a relationship set.
- Example: Student is enrolled in a Course

Binary Relationship Set

- **Ternary relationship set** is a relationship set where three entity sets participate in a relationship set.

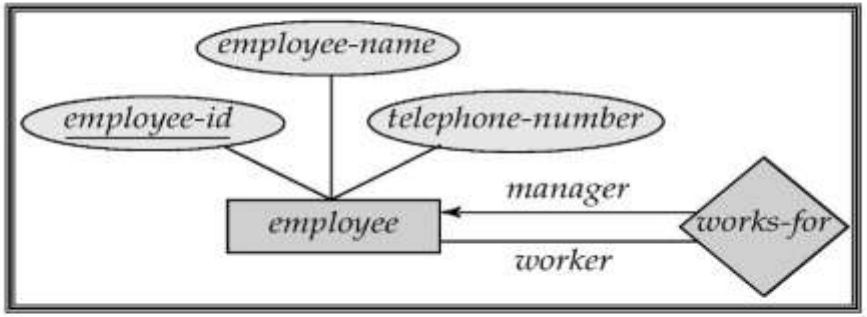


Ternary Relationship Set

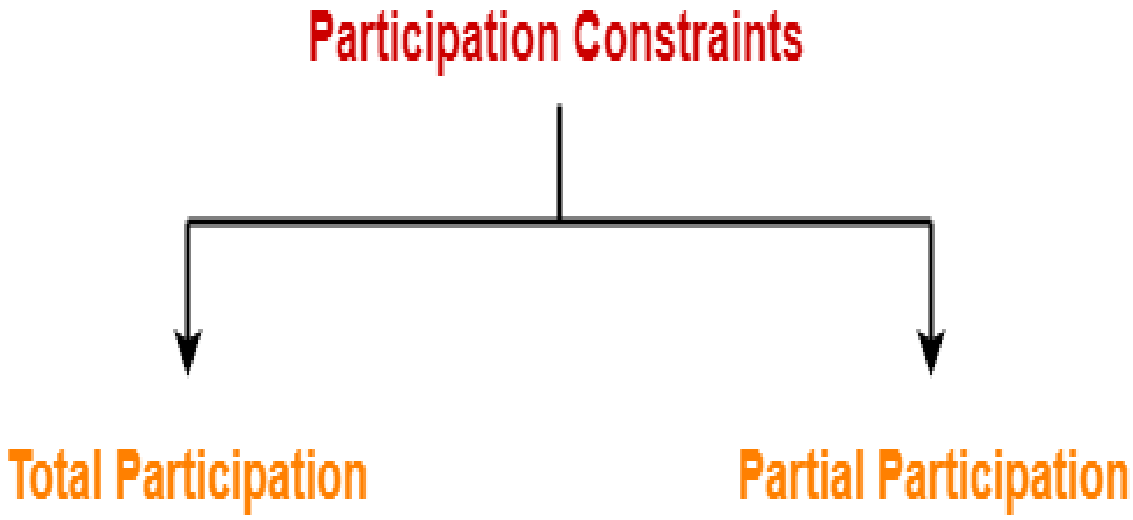
- **N-ary relationship set** is a relationship set where ‘n’ entity sets participate in a relationship set.

- **Roles**
  - Roles are indicated in E-R diagrams by labeling the lines that connect diamonds to rectangles.
  - Role labels are optional, and are used to clarify semantics of the relationship
  - The labels “manager” and “worker” are called roles; they specify how employee entities interact via the works-for relationship set

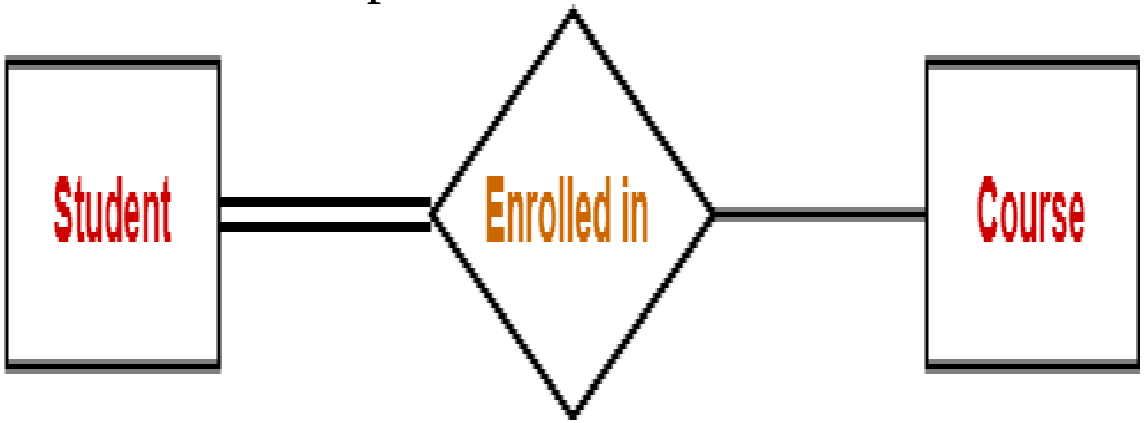
Roles in ER diagrams



- **Participation constraints** define the least number of relationship instances in which an entity must compulsorily participate.

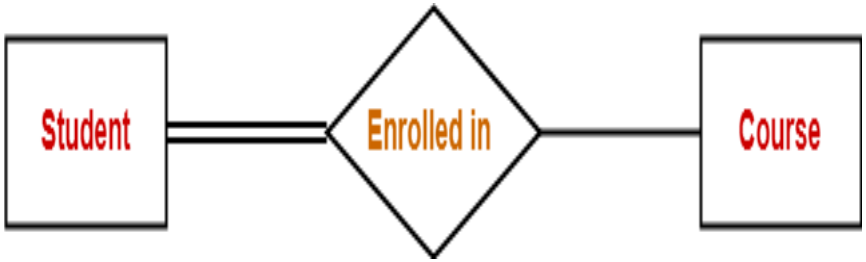


- Total Participation
  - It specifies that each entity in the entity set must compulsorily participate in at least one relationship instance in that relationship set.
  - That is why, it is also called as **mandatory participation.**
  - Total participation is represented using a **double line** between the entity set and relationship set.
- Example



- Here,
  - Double line between the entity set “Student” and relationship set “Enrolled in” signifies total participation.
  - It specifies that each student must be enrolled in at least one course.

- **Partial Participation**
- It specifies that each entity in the entity set may or may not participate in the relationship instance in that relationship set.
- That is why, it is also called as **optional participation.**
- Partial participation is represented using a single line between the entity set and relationship set
- Example

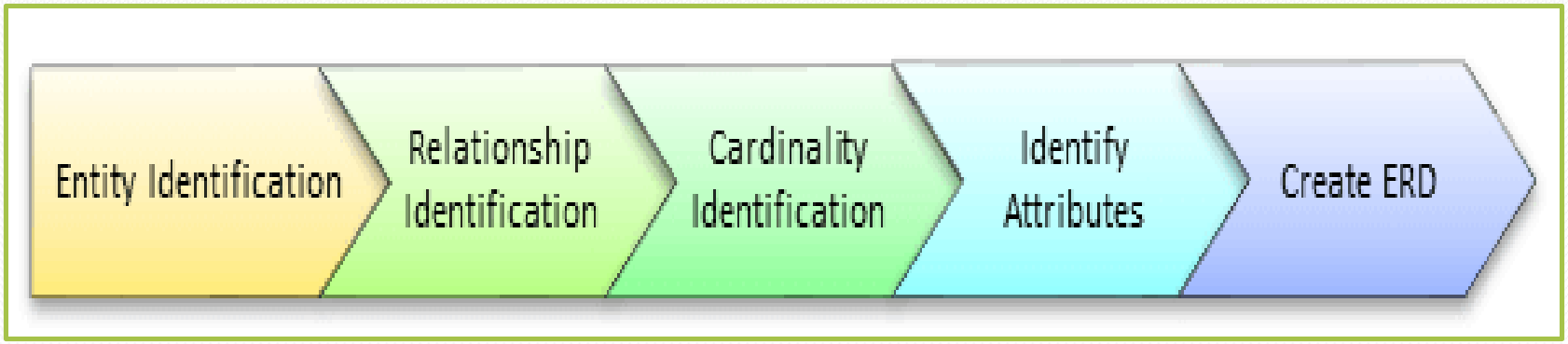


- Here,
- Single line between the entity set “Course” and relationship set “Enrolled in” signifies partial participation.
- It specifies that there might exist some courses for which no enrollments are made.

Relationship between Cardinality and Participation Constraints

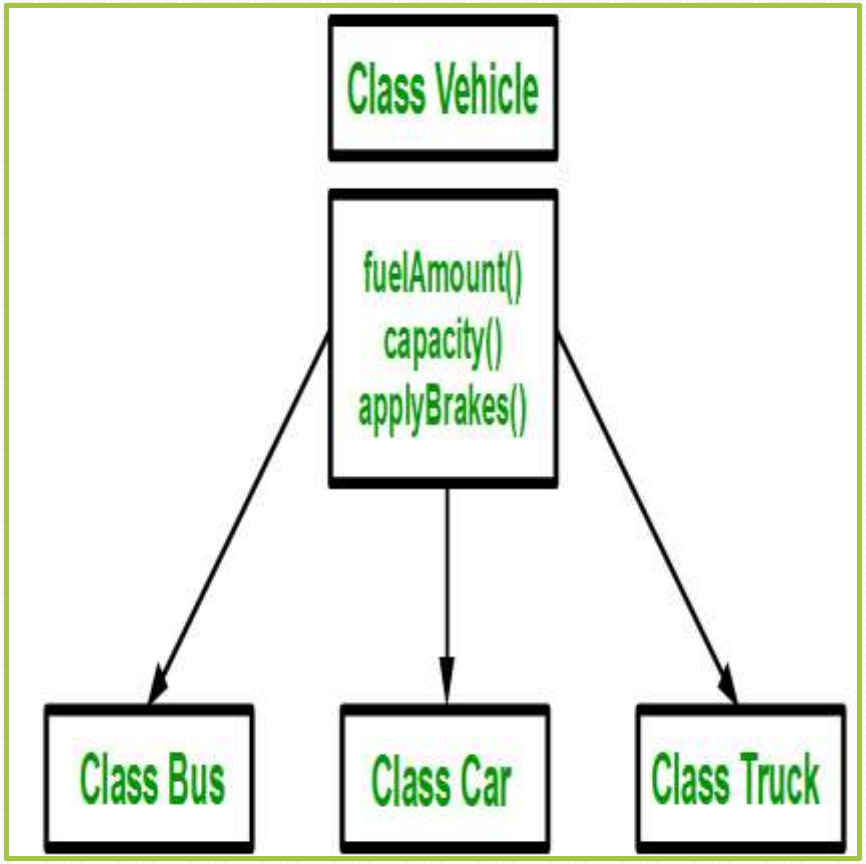
- Minimum cardinality tells whether the participation is partial or total.
  - If minimum cardinality = 0, then it signifies partial participation.
  - If minimum cardinality = 1, then it signifies total participation.
- Maximum cardinality tells the maximum number of entities that participates in a relationship set.

Steps to Create an ERD

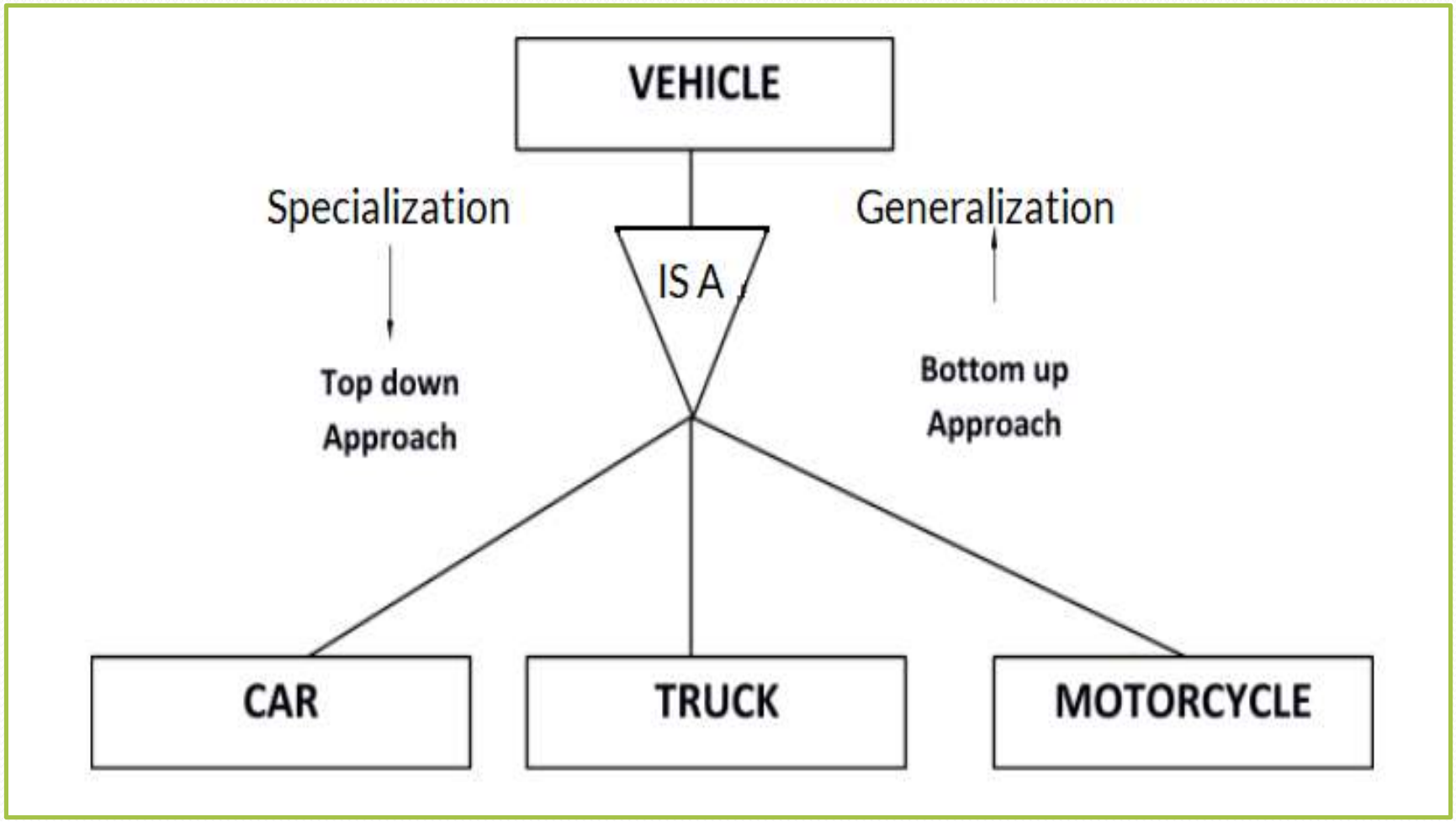


Inheritance

- Mechanism in which one class acquires the property of another class.
- For example, a child inherits the traits of his/her parents.
- With inheritance, we can reuse the fields and methods of the existing class.
- Inheritance facilitates **Reusability**



Subclass & Superclass

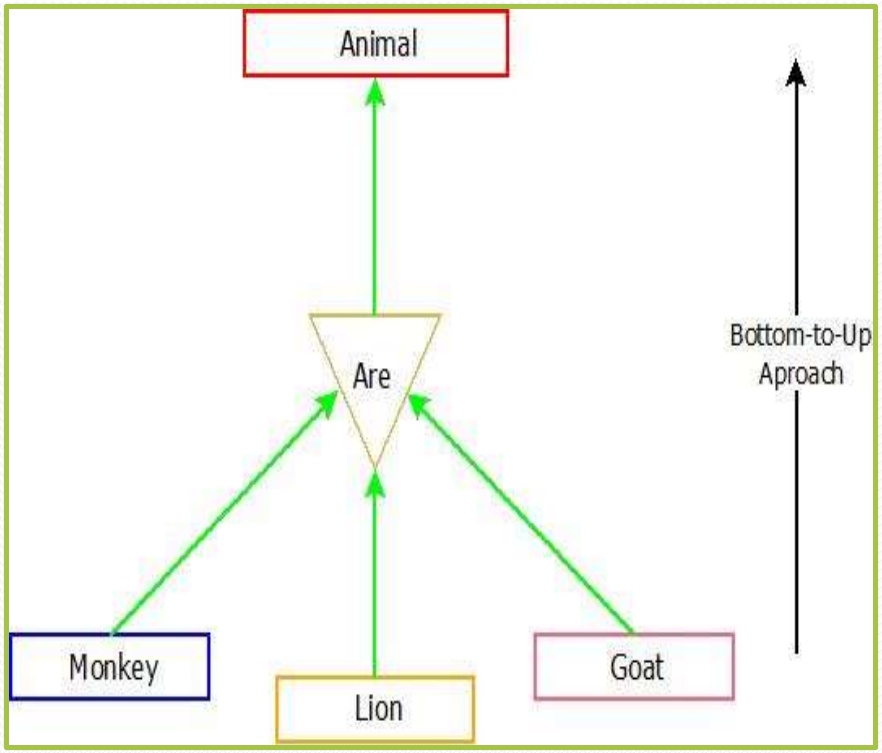


- **Subclass:**
  - Inherits all attributes of the **Super Class**
  - Inherits all relationships of the **Super Class**
  - Usually has its own attributes or relationships
  - Is drawn within the **Super Class**
- **Super Class:**
  - A **Super Class is a generic entity type that has a relationship with one or** more **sub class.**
  - A super class is the class from which many subclasses can be created.
  - The subclasses inherit the characteristics of a super class. The super class is also known as the parent class or base class.

- Example :
  - **PERSON**: Name, Address, Birthdate, Age, SSN
  - **EMPLOYEE**: Name, Address, Birthdate, Age, SSN, Salary, HireDate, Seniority
  - **STUDENT**: Name, Address, Birthdate, Age, SSN, Major,GPA
- OR:
  - **EMPLOYEE subtype-of PERSON: Salary, HireDate,** Seniority
  - **STUDENT subtype-of PERSON: Major, GPA**

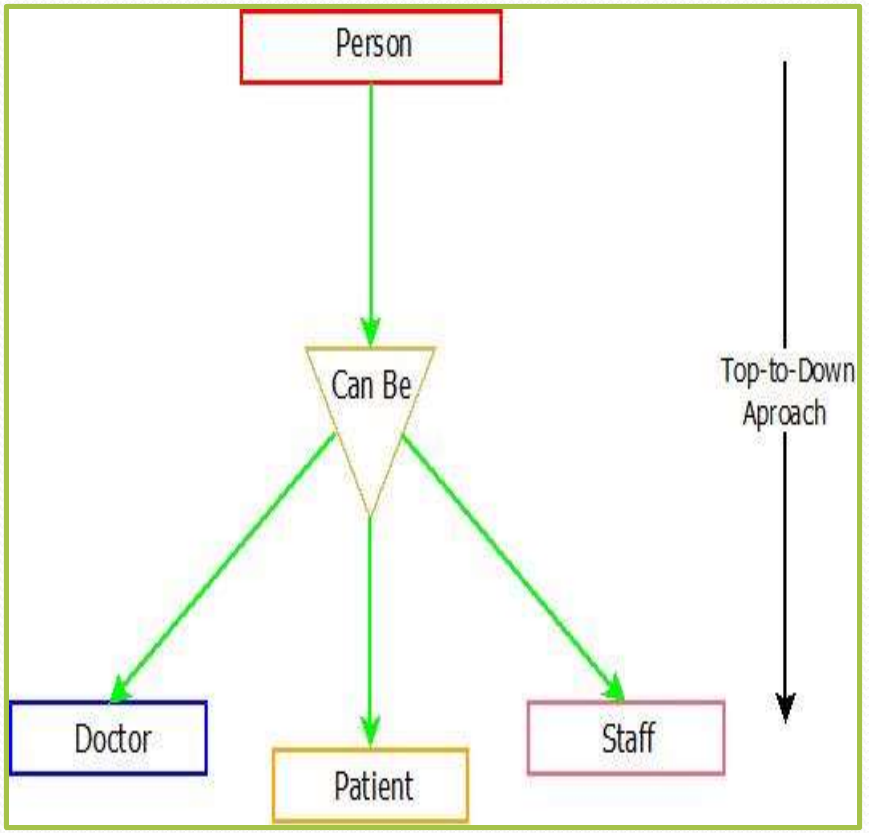
Generalization

- **Generalization** is a **bottom-up approach** in which two lower level entities combine to form a higher level entity.
- Generalization is the process of extracting common properties from a set of entities and create a generalized entity from it.



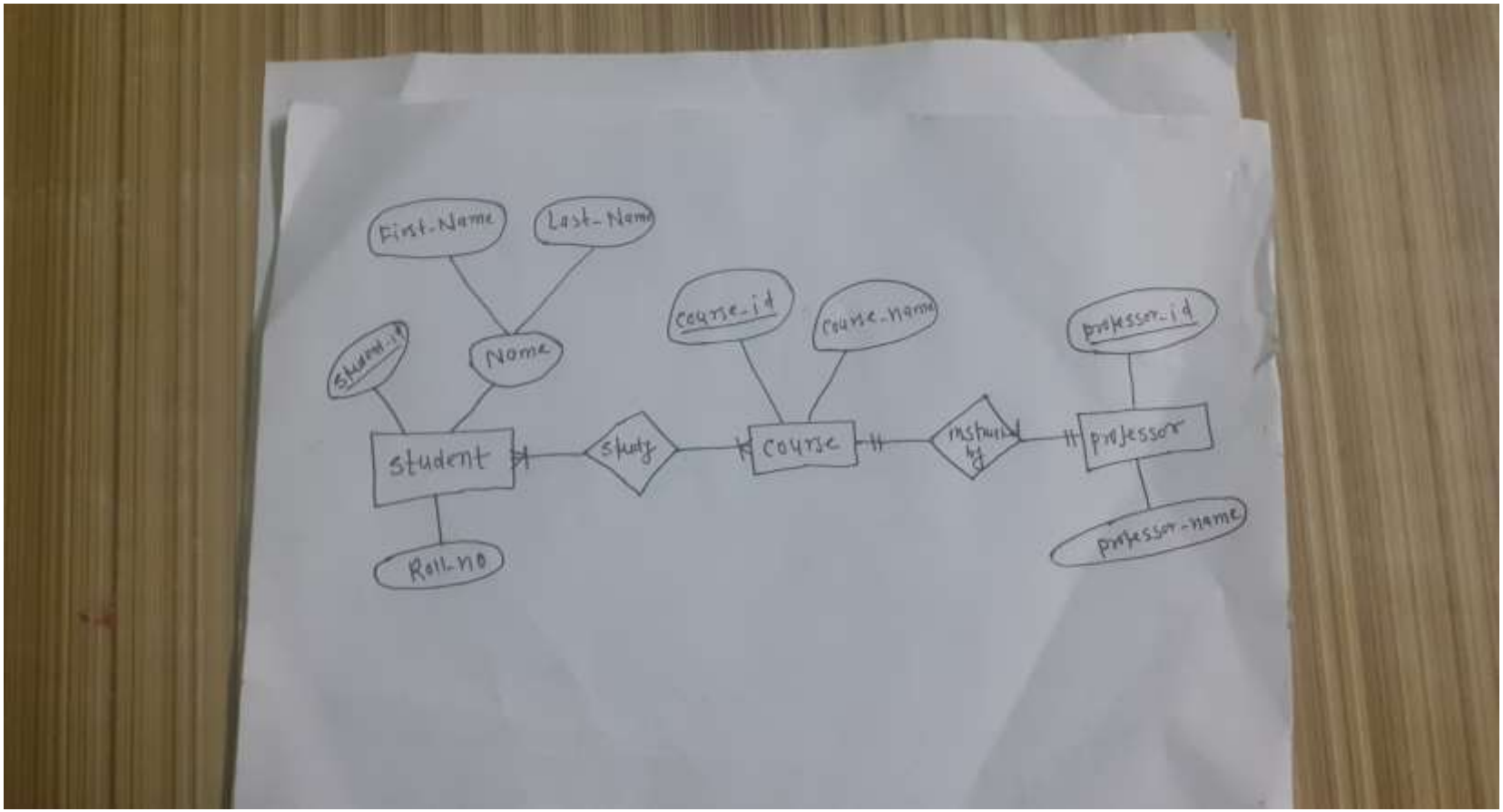
Specialization

- 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.

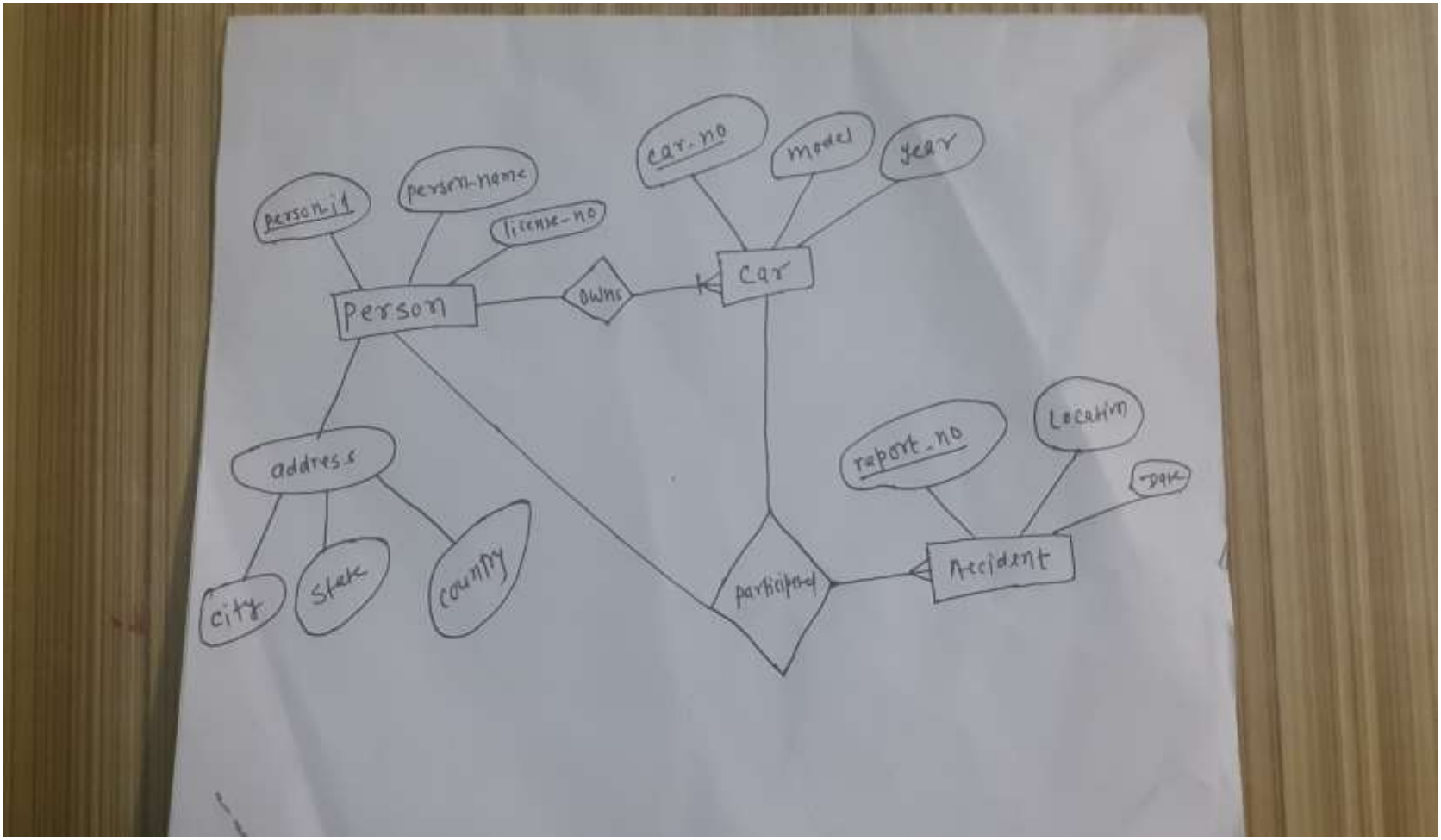


- In a university, a Student enrolls in Courses. A student must be assigned to at least one or more Courses. Each course is taught by a single Professor. To maintain instruction quality, a Professor can deliver only one course
- Construct an E-R diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents
- Consider a database used to record the marks that students get in different exams of different course offerings.
  - Construct an E-R diagram that models exams as entities, and uses a ternary relationship, for the above database.
- Design an E-R diagram for keeping track of the exploits of your favorite sports team. You should store the matches played, the scores in each match, the players in each match and individual player statistics for each match. Summary statistics should be modeled as derived attributes.

- In a university, a Student enrolls in Courses. A student must be assigned to at least one or more Courses. Each course is taught by a single Professor. To maintain instruction quality, a Professor can deliver only one course



- Construct an E-R diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents



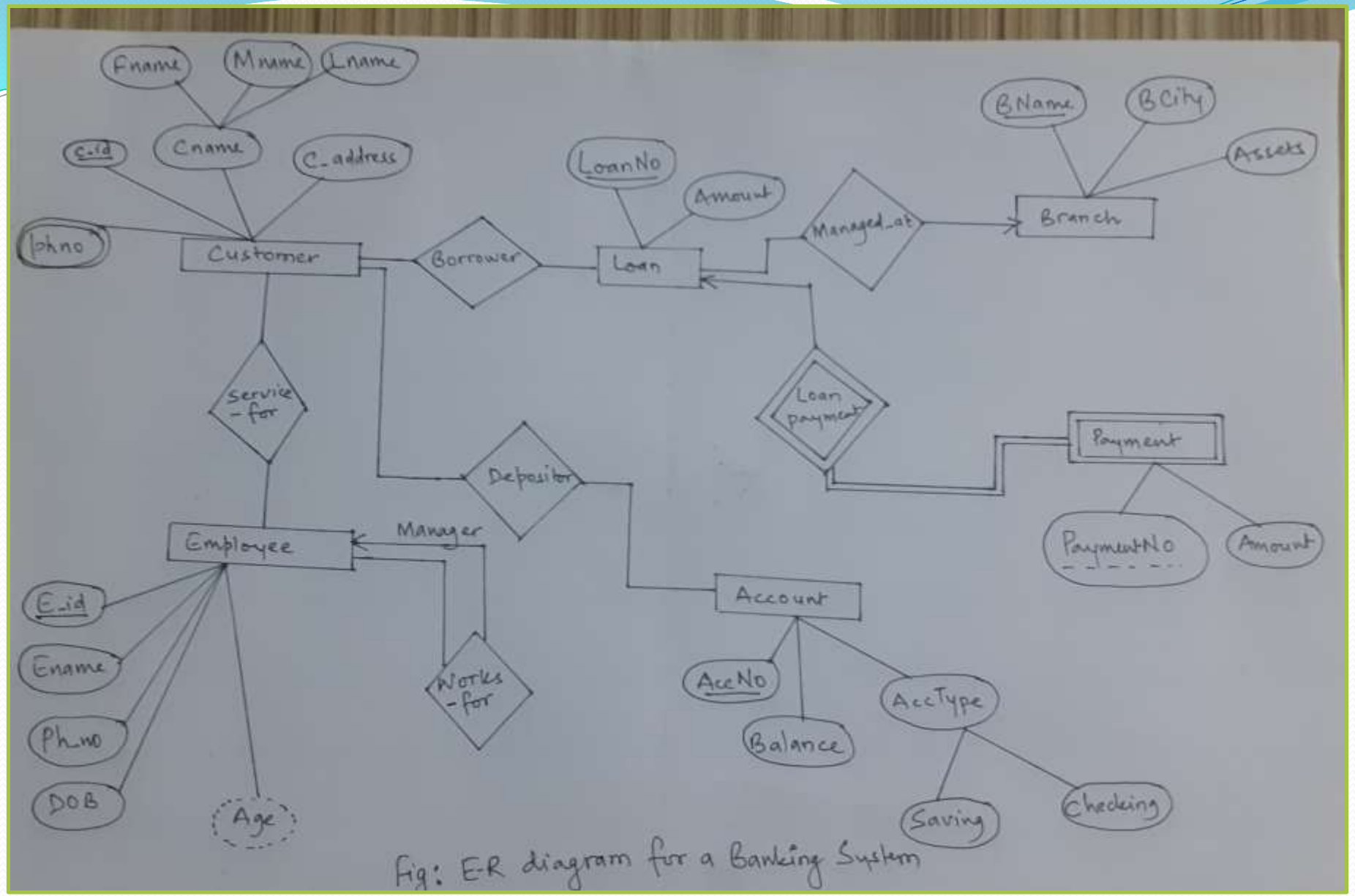


Fig: E-R diagram for a Banking System

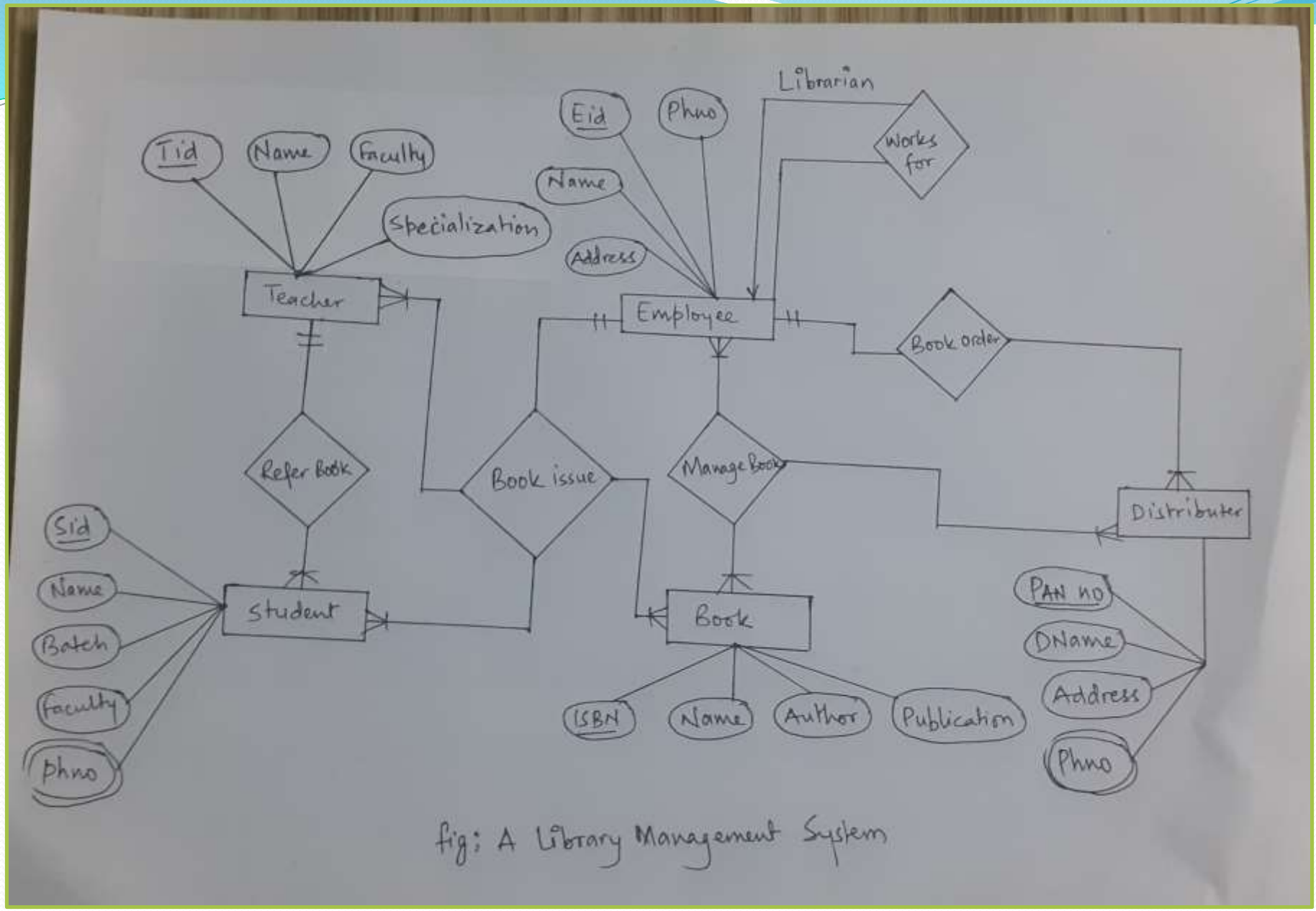
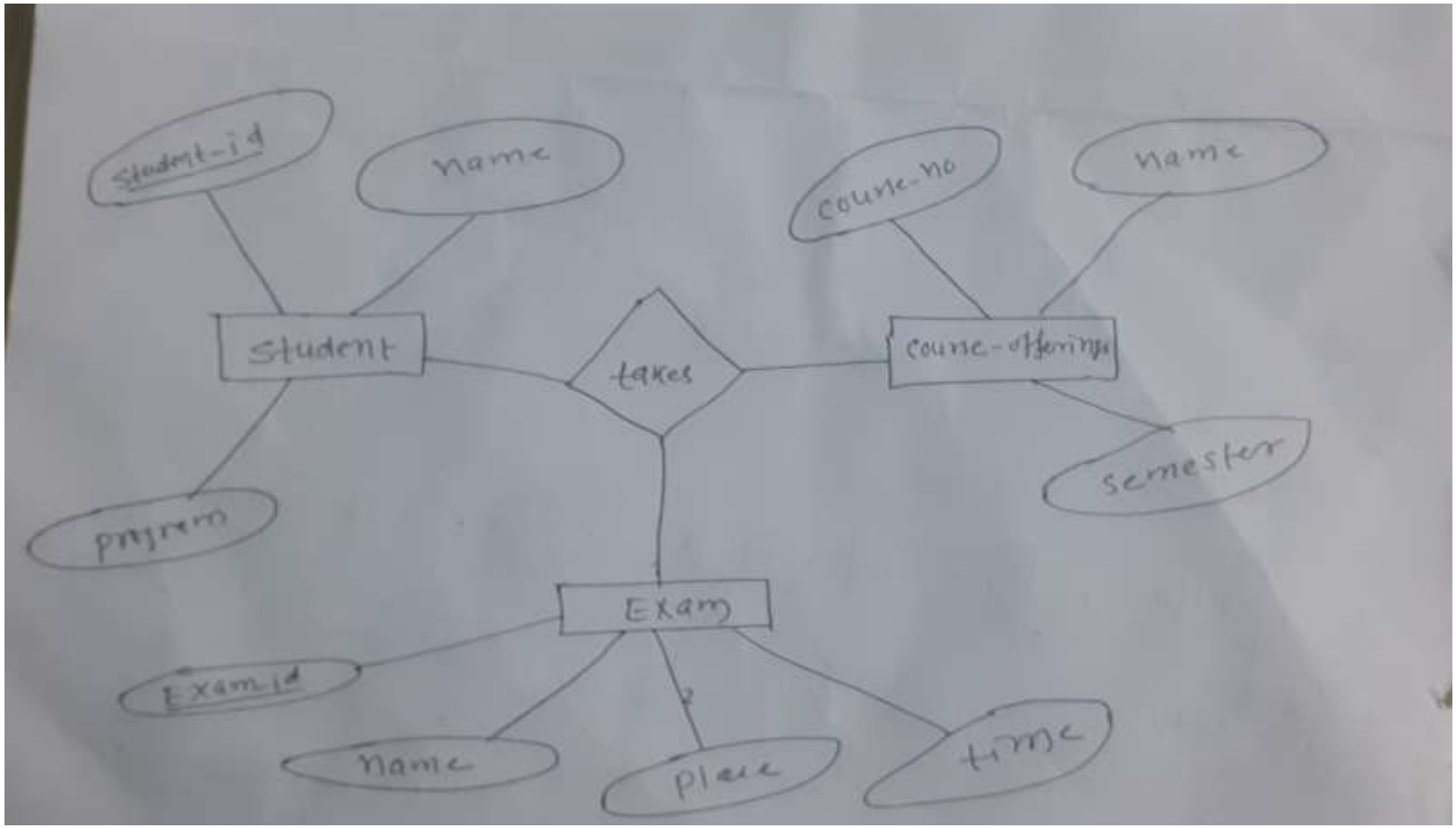
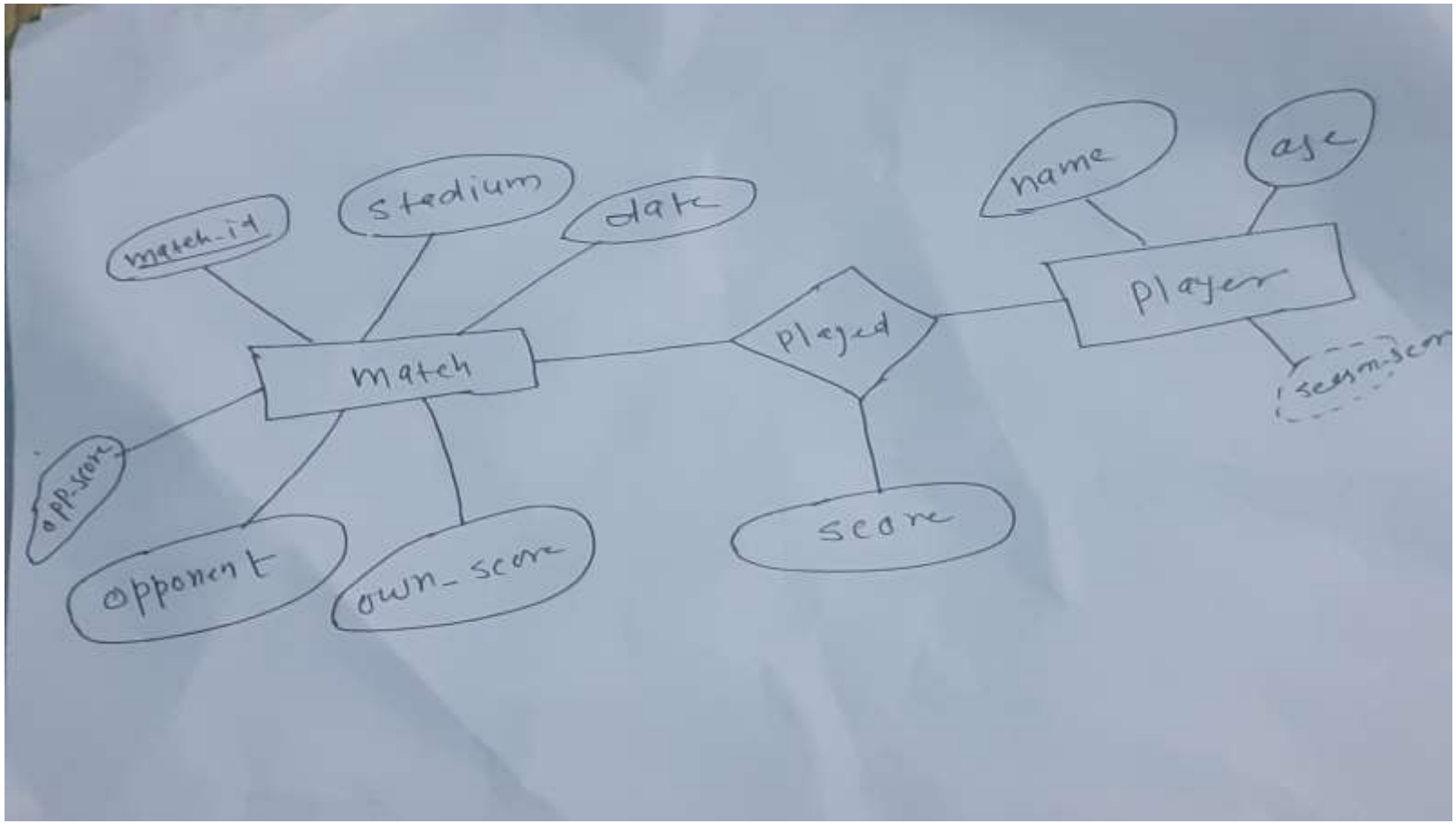


fig: A Library Management System

- Consider a database used to record the marks that students get in different exams of different course offerings.
  - Construct an E-R diagram that models exams as entities, and uses a ternary relationship, for the above database.



- Design an E-R diagram for keeping track of the exploits of your favorite sports team. You should store the matches played, the scores in each match, the players in each match and individual player statistics for each match. Summary statistics should be modeled as derived attributes



1. Company is organized into departments ﬁDepartment controls a number of projects. Employee works for department. ﬁEmployee: store each employee's name, Social Security number, address, salary, gender, and birth date. Also keep track of the dependents of each employee.
2. ER diagram of Bank has the following description :
   - Bank have Customer.
   - Banks are identified by a name, code, address of main office.
   - Banks have branches.
   - Branches are identified by a branch_no., branch_name, address.
   - Customers are identified by name, cust-id, phone number, address.
   - Customer can have one or more accounts.
   - Accounts are identified by acc_no., acc_type, balance.
   - Customer can avail loans.
   - Loans are identified by loan_id, loan_type and amount.
   - Account and loans are related to bank's branch.

3. Create an ER diagram for a ecommerce website in which customer can place zero or many orders. Orders contains one or many products.

4. Consider a university database for the scheduling of classrooms for -final exams. Consider a following entities:

- Course with attributes name, department, and c-number
- Section with attributes s-number and enrollment
- Room with attributes r-number, capacity, and building

Show an E-R diagram illustrating the use of all three additional entity sets listed

5. Construct an ER-Diagram for the following NFL database

You are given the requirement for a simple database for the National Football League(NFL) .The NFL has many teams, and each team has a name , a city ,a coach ,a captain and a set of players. Each player belongs to only one team and each and each player has a name ,a position (such as a left wing, midfielder, goalkeeper) a skill level and a set of injury records. A team captain is also a player and a game is played between two teams(referred as a host team and guest team) and has a match date (Such as October 11,,2020) and score (such as 2 to 5)

6. Design an E-R diagram for a database for an airlines system. The database must keep track of customers and their reservations, flights and their status' seat assignments on individual flights and the schedule and routing of future flights. Apply all the database design constraints as much as possible .

7. Draw the Entity : Relationship Diagram (ERD) with appropriate mapping cardinalities for the following scenario.

   Patients are treated in a single ward by the the doctors assigned to them. Healthcare assistants also attend to the patients; a number of these are associated with each ward. Each patient is required to take a variety of drugs a certain number of times per day and for varying lengths of time. The system must record details concerning patient treatment and staff payment. Some staffs are paid part time and doctors and healthcare assistants work varying amounts of overtime at varying rates, the system will also need to track what treatments are required for which patients.