01CT0407 - Database Management System

Unit - 2

Data Models

E-R Diagram

Prof. Harikesh Chauhan
Information Communication and Technology

Outline

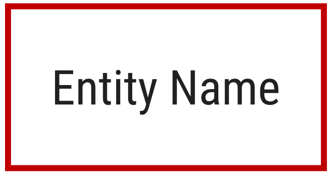
- Basic concept of E-R diagram
- Types of Attributes
- Mapping Cardinality
- Weak Entity Sets
- Extended E-R features
- Generalization and Specialization
- Constraints on Specialization and Generalization
- Aggregation
- E-R diagram of Hospital Management System
- Reduction to E-R Database Schema
- Integrity Constraints

Basic concepts

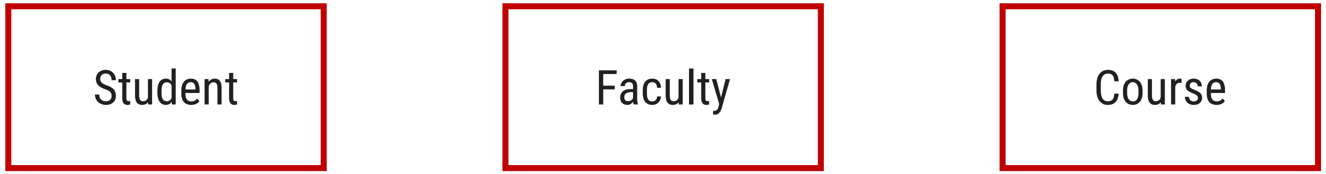
- What is Database Design?
  - Database Design is a collection of processes that facilitate the **designing**, **development**, **implementation** and **maintenance** of enterprise database management systems.
- What is E-R diagram?
  - E-R diagram: (Entity-Relationship diagram)
  - It is **graphical (pictorial) representation** of database.
  - It uses different types of symbols to represent different objects of database.

Entity

- An entity is a **person**, a **place** or an **object**.
- An entity is represented by a **rectangle** which contains the name of an entity.
- Entities of a college database are:
  - Student
  - Professor/Faculty
  - Course
  - Department
  - Result
  - Class
  - Subject



Symbol



Exercise Write down the different entities of bank database.

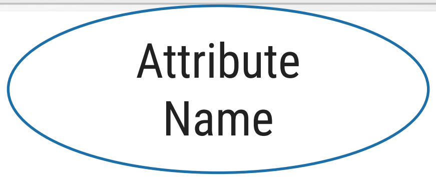
Exercise Write down the different entities of hospital database.

Entity Set

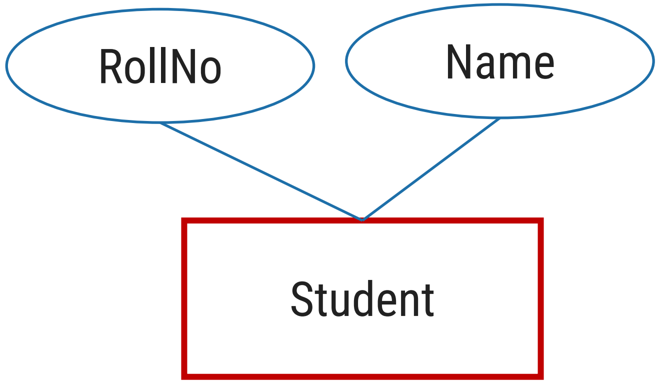
- It is a **set (group) of entities** of **same type**.
- Examples:
  - All persons having an account in a bank
  - All the students studying in a college
  - All the professors working in a college
  - Set of all accounts in a bank

Attributes

- Attribute is **properties** or details about an entity.
- An attribute is represented by an **oval** containing name of an attribute.
- Attributes of Student are:
    - Roll No
    - Student Name
    - Branch
    - Semester
    - Address
    - Mobile No
    - Age
    - SPI
    - Backlogs



Symbol



Exercise Write down the different attributes of Faculty entity.

Exercise Write down the different attributes of Account entity.

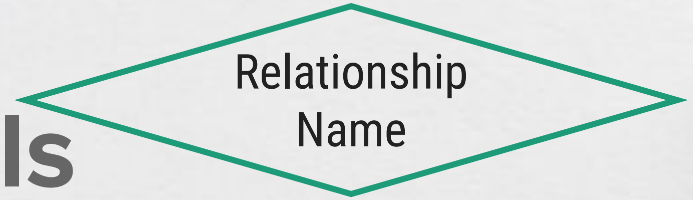
Relationship

01CT0407 - Database Management System

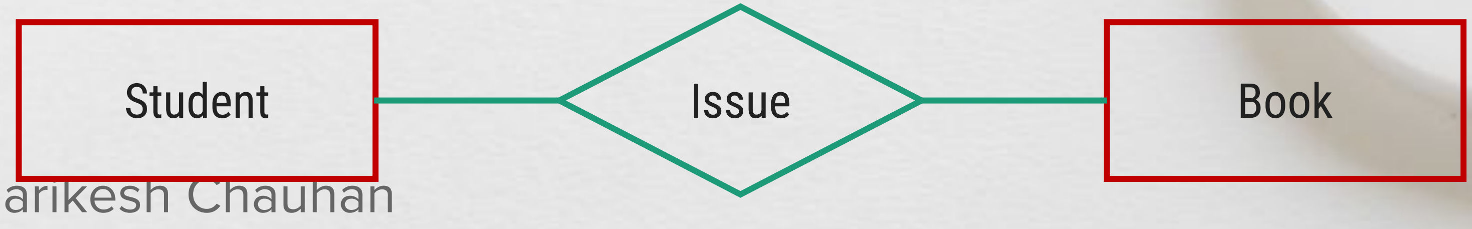
Unit - 2

Data Models

E-R Diagram

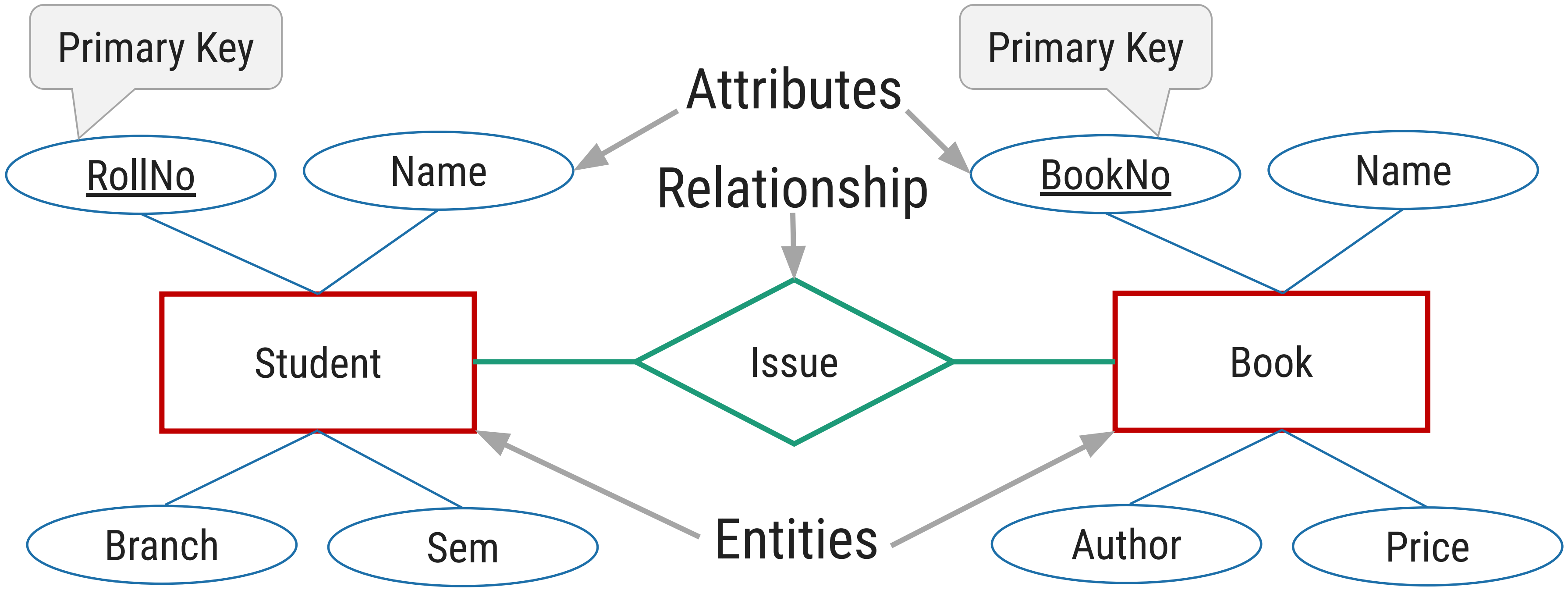


Symbol



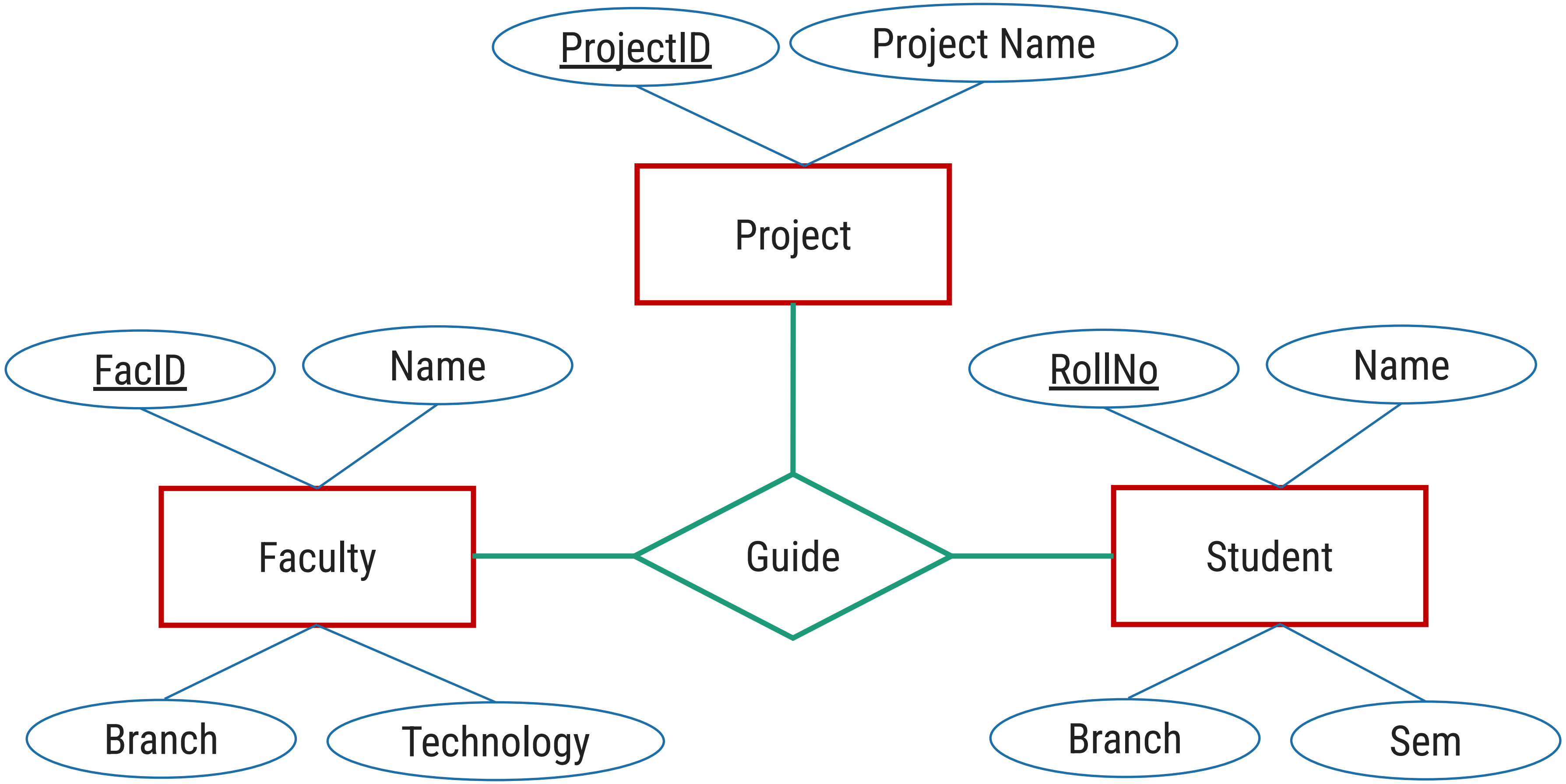
Prof. Harikesh Chauhan
Information Communication and Technology

E-R Diagram of a Library System



Each and every entity must have one primary key attribute.
Relationship between 2 entities is called binary relationship.

Ternary Relationship

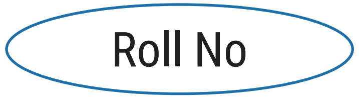
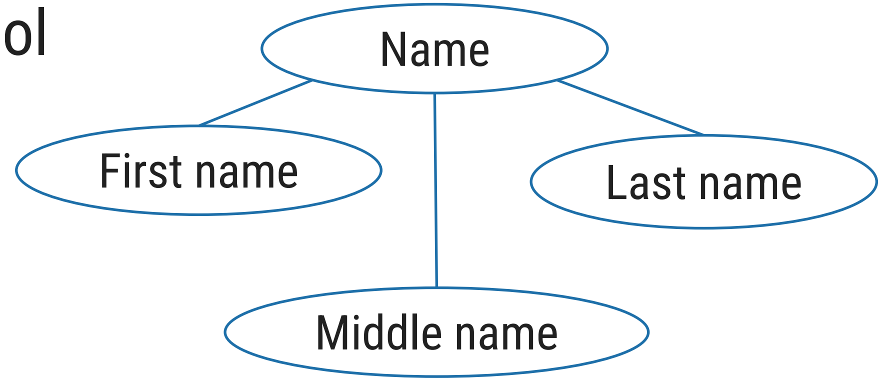


Relationship between 3 entities is called ternary relationship.

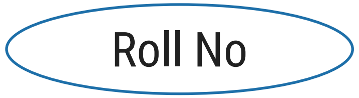
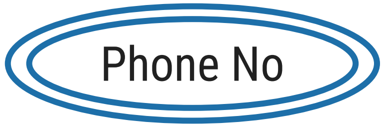
Exercise

- Draw an E-R diagram of following pair of entities
  - Customer & Account
  - Customer & Loan
  - Doctor & Patient
  - Student & Project
  - Student & Teacher
    - Note: Take four attributes per entity with one primary key attribute.
      Keep proper relationship between two entities.

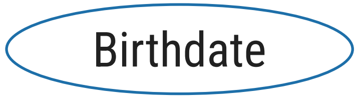
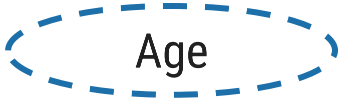
Types of Attributes

Simple Attribute	Composite Attribute
Cannot be divided into subparts	Can be divided into subparts
E.g. RollNo, CPI	E.g. Name (first name, middle name, last name) Address (street, road, city)
Symbol  	Symbol 

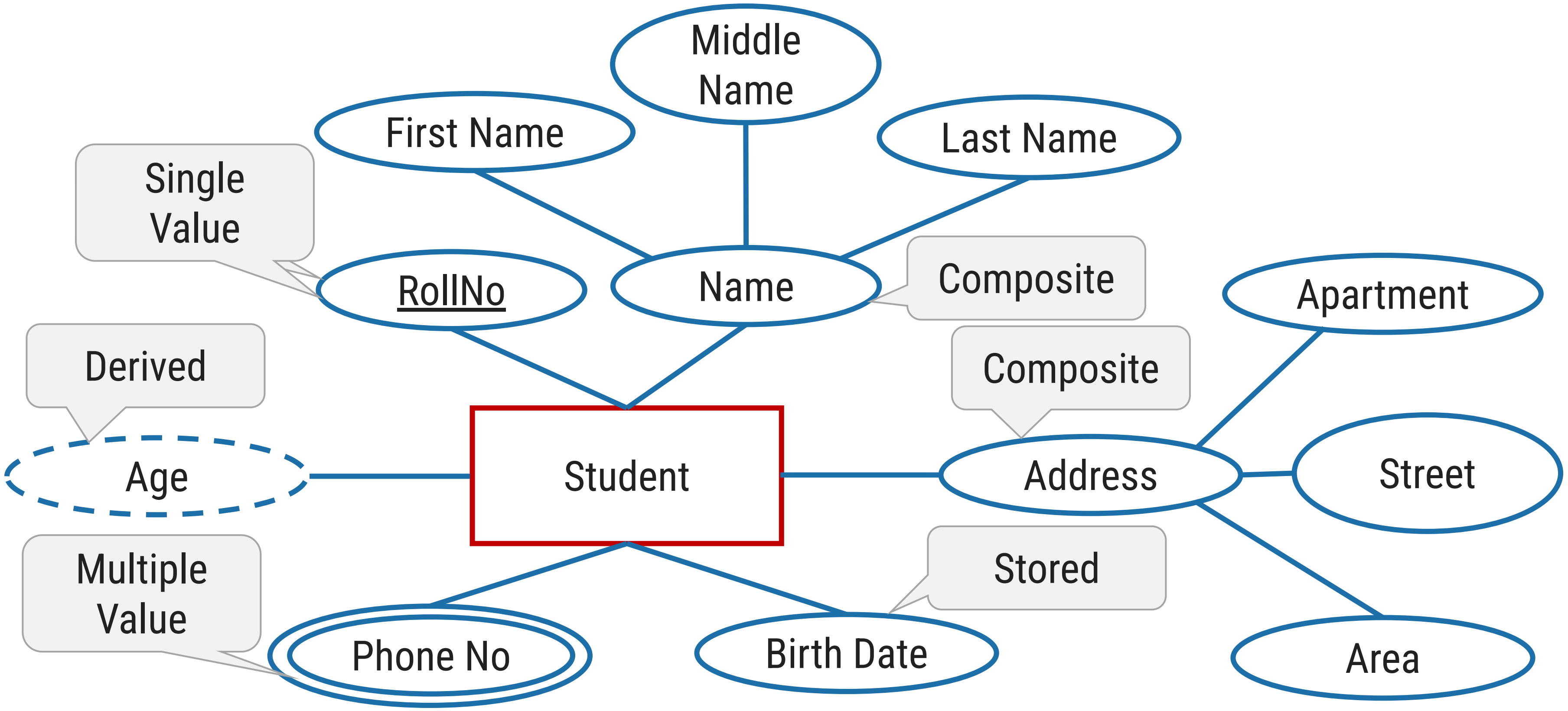
Types of Attributes

Single-valued Attribute	Multi-valued Attribute
Has single value	Has multiple (more than one) value
E.g. RollNo, CPI	E.g. PhoneNo (person may have multiple phone nos) EmailID (person may have multiple emails)
Symbol 	Symbol 

Types of Attributes

Stored Attribute	Derived Attribute
It’s value is stored manually in database	It’s value is derived or calculated from other attributes
E.g. Birthdate	E.g. Age (can be calculated using current date and birthdate)
Symbol 	Symbol 

Entity with all types of Attributes

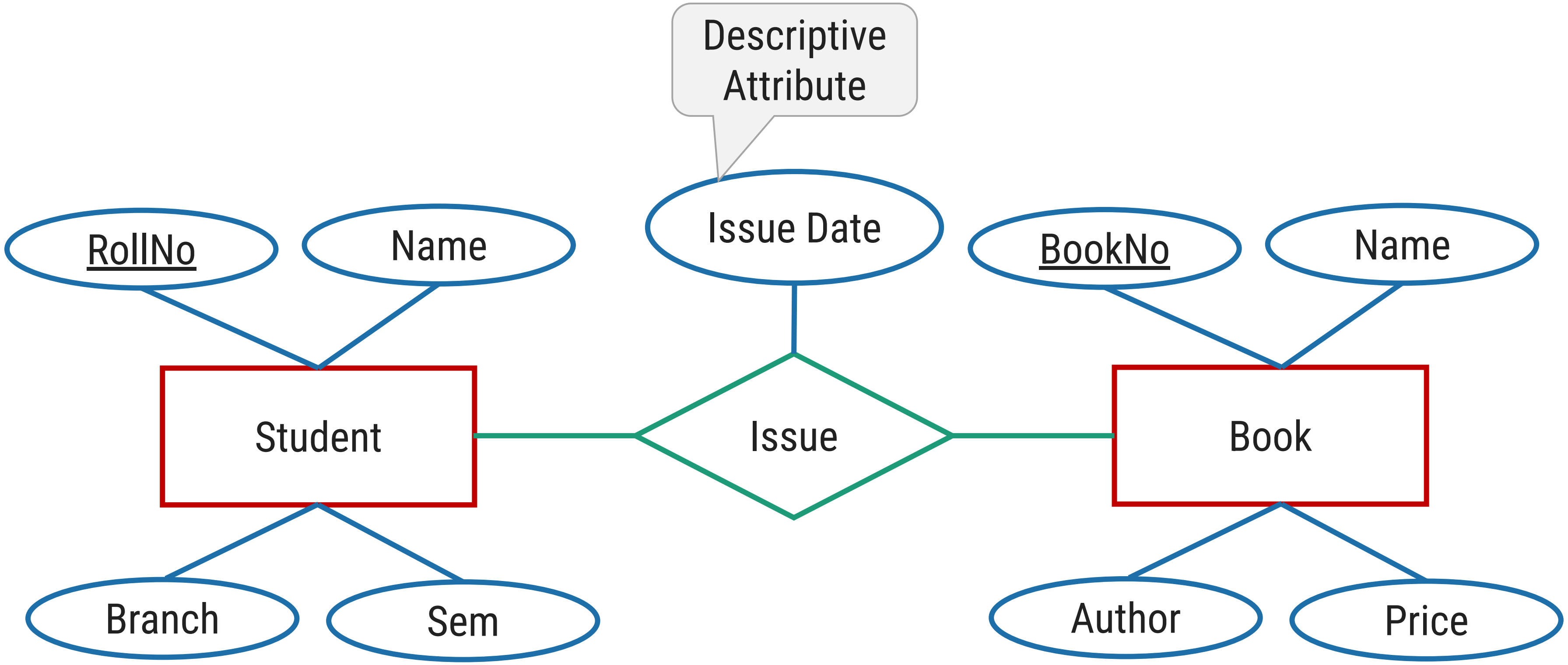


Exercise

- Draw an E-R diagram of Banking Management System.
- Draw an E-R diagram of Hospital Management System.
- Draw an E-R diagram of College Management System.
    - Take only 2 entities
    - Keep proper relationship between two entities
    - Use all types of attributes

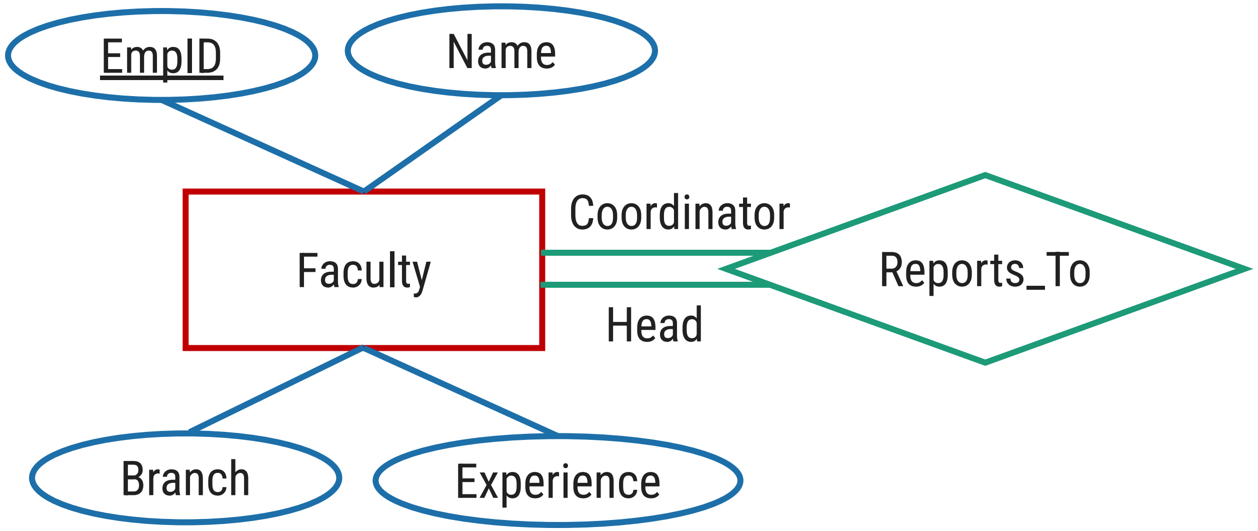
Descriptive Attribute

- **Attributes of the relationship** is called descriptive attribute.



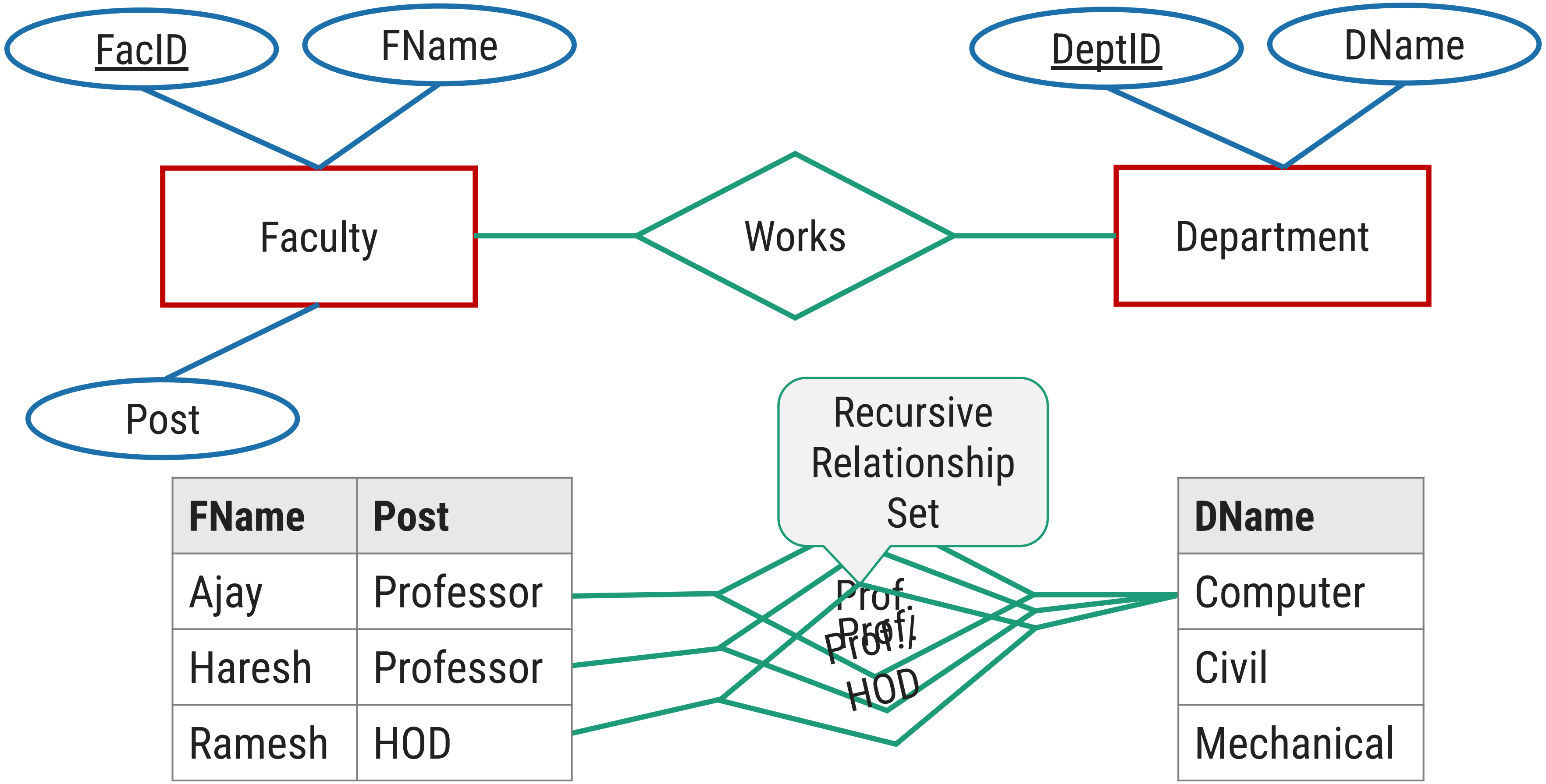
Role

- Roles are indicated by labeling the lines that connect diamonds (relationship) to rectangles (entity).
- The labels “Coordinator” and “Head” are called roles; it specify the function that an entity plays in a relationship.
- Role labels are optional, and are used to clarify semantics (meaning) of the relationship.



Recursive Relationship Set

- The same **entity participates in a relationship set more than once** then it is called recursive relationship set.



FName	Post
Ajay	Professor
Haresh	Professor
Ramesh	HOD

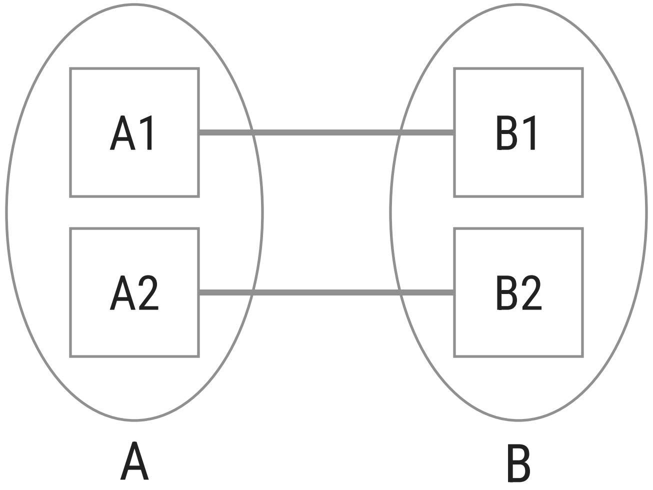
DName
Computer
Civil
Mechanical

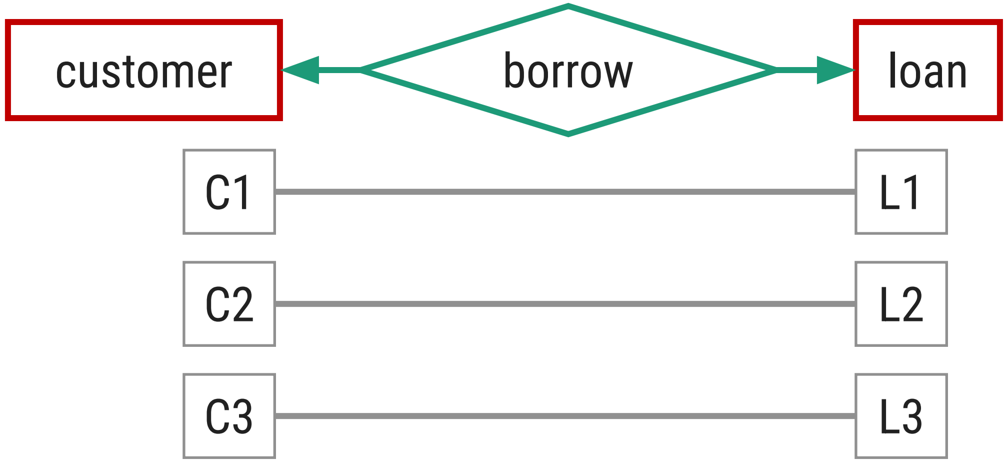
Mapping Cardinality (Cardinality Constraints)

- It represents the **number of entities of another entity set** which are **connected to an entity** using a relationship set.
- It is most **useful in describing binary relationship sets**.
- For a binary relationship set the mapping cardinality must be one of the following types:
  - One to One
  - One to Many
  - Many to One
  - Many to Many

One-to-One relationship (1 – 1)

- An entity in **A is associated with only one entity in B** and an entity in **B is associated with only one entity in A**.

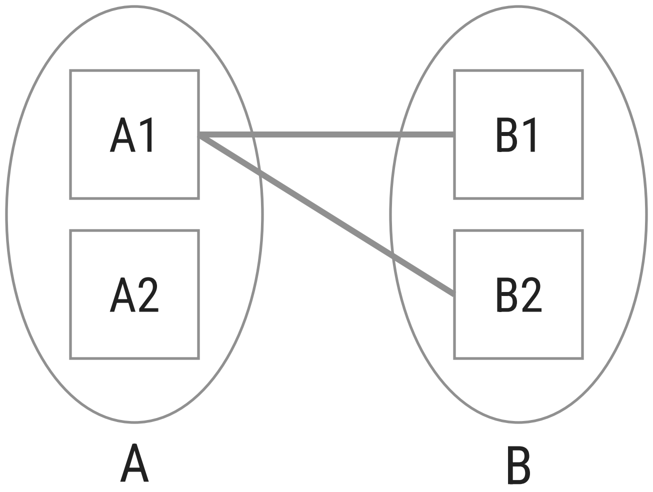


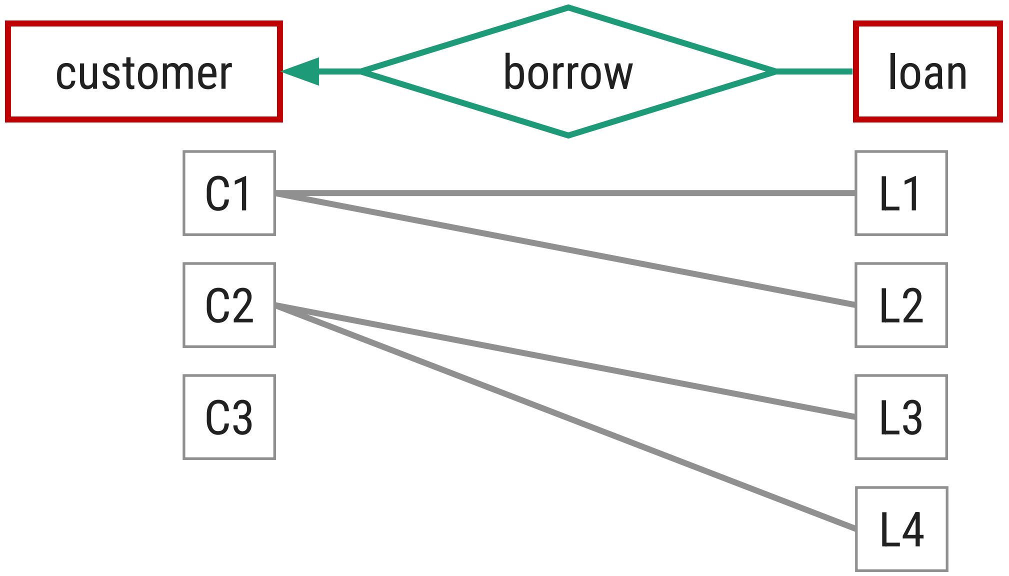


- Example: A **customer is connected with only one loan** using the relationship borrower and a **loan is connected with only one customer** using borrower.

One-to-Many relationship (1 – N)

- An entity in **A is associated with more than one entities in B** and an entity in **B is associated with only one entity in A**.

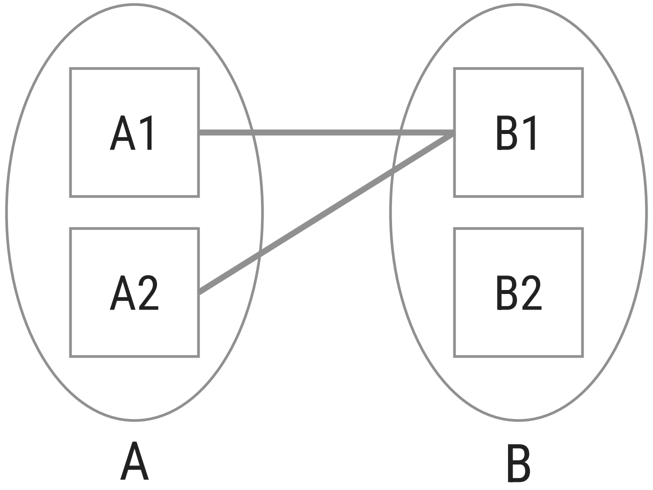


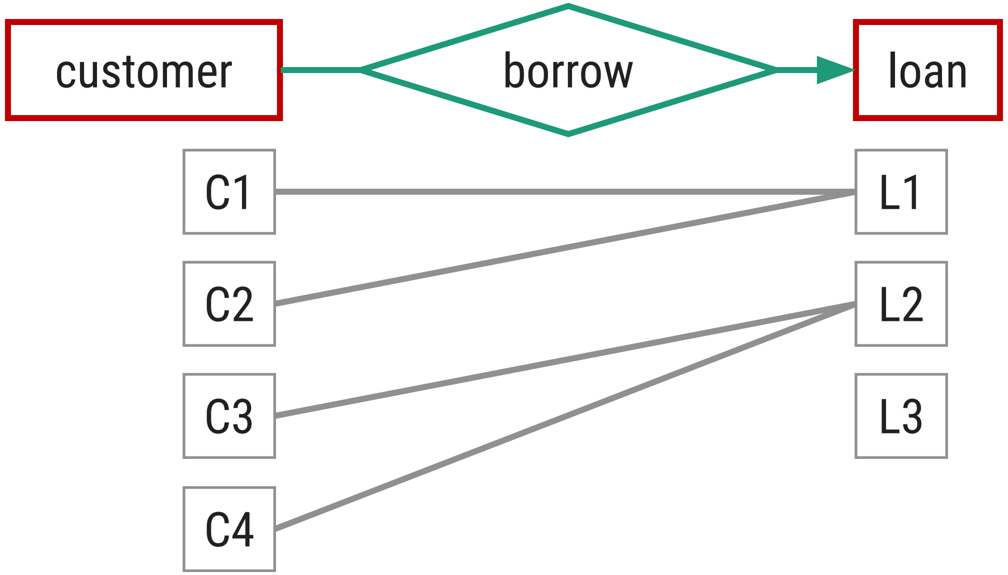


- Example: A **loan is connected with only one customer** using borrower and a **customer is connected with more than one loans using borrower**.

Many-to-One relationship (N – 1)

- An entity in **A is associated with only one entity in B** and an entity in **B is associated with more than one entities in A**.

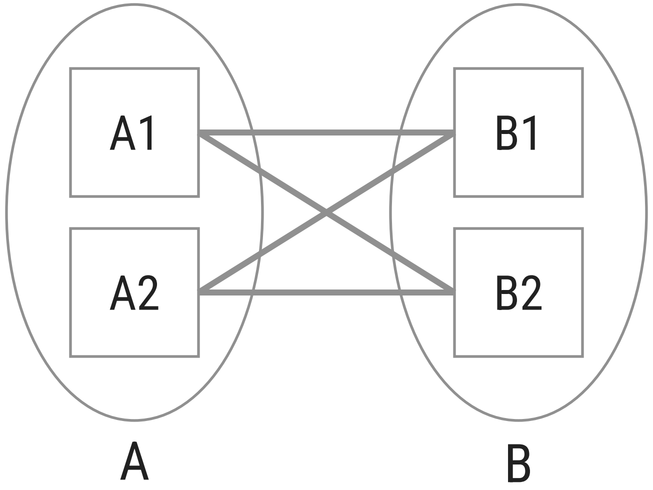


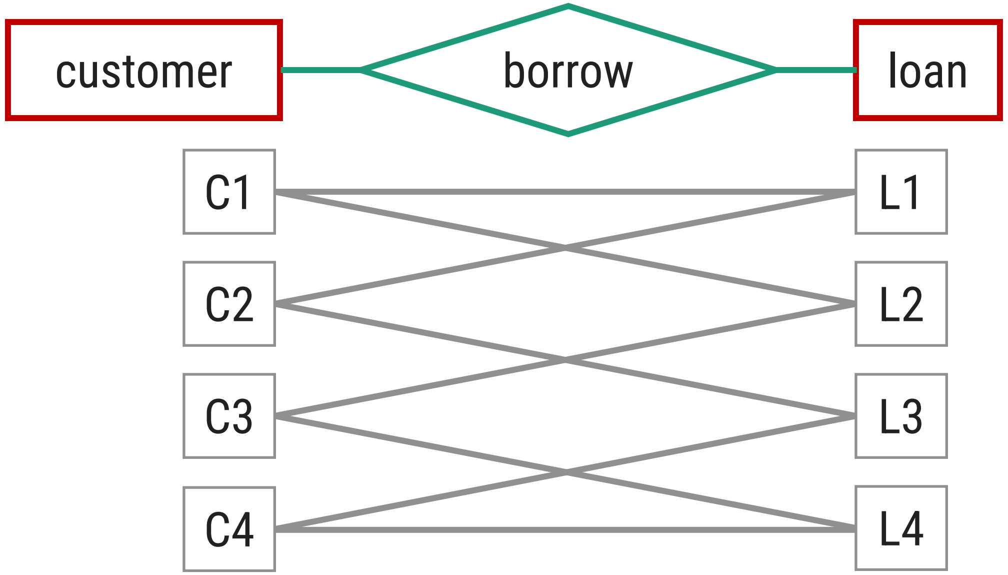


- Example: A **loan is connected with more than one customer** using borrower and a **customer is connected with only one loan** using borrower.

Many-to-Many relationship (N – N)

- An entity in **A is associated with more than one entities in B** and an entity in **B is associated with more than one entities in A**.





- Example: A **customer is connected with more than one loan** using borrower and a **loan is connected with more than one customer** using borrower.

Mapping Cardinality (Cardinality Constraints) [Exercise]

- Draw an E-R diagram and specify which type of mapping cardinality will be there in the following examples:
  - Each customer has only one account in the bank and each account is held by only one customer. [single account]
  - Each customer has only one account in the bank but an account can be held by more than one customer. [joint account]
  - A customer may have more than one account in the bank but each account is held by only one customer. [multiple accounts]
  - A customer may have more than one account in the bank and each account is held by more than one customer. [join account as well as multiple accounts]
  - A student can work in more than one project and a project can be done by more than one student.
  - A student can issue more than one book but a book is issued to only one student.
  - A subject is taught by more than one faculty and a faculty can teach more than one subject.

Exercise

- Draw an E-R diagram of Banking Management System.
- Draw an E-R diagram of Hospital Management System.
- Draw an E-R diagram of College Management System.
  - Take only 3 to 4 entities
  - Keep proper relationship between two entities
  - Use all types of attributes
  - Use Mapping Cardinality

Participation Constraints

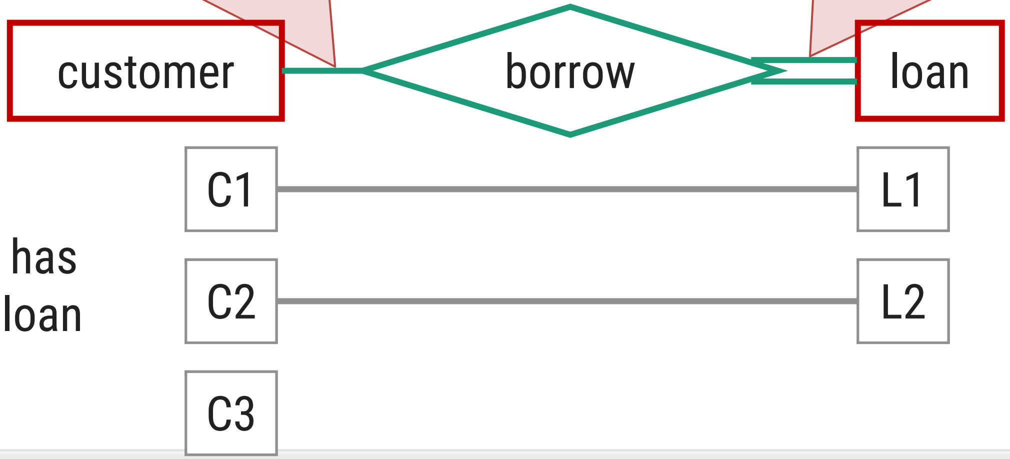
- It specifies the **participation of an entity set** in a relationship set.
- There are two types participation constraints
  - Total participation
  - Partial participation

Partial participation

- some entities in the entity set may not participate in any relationship in the relationship set.
- indicated by single line

Total participation

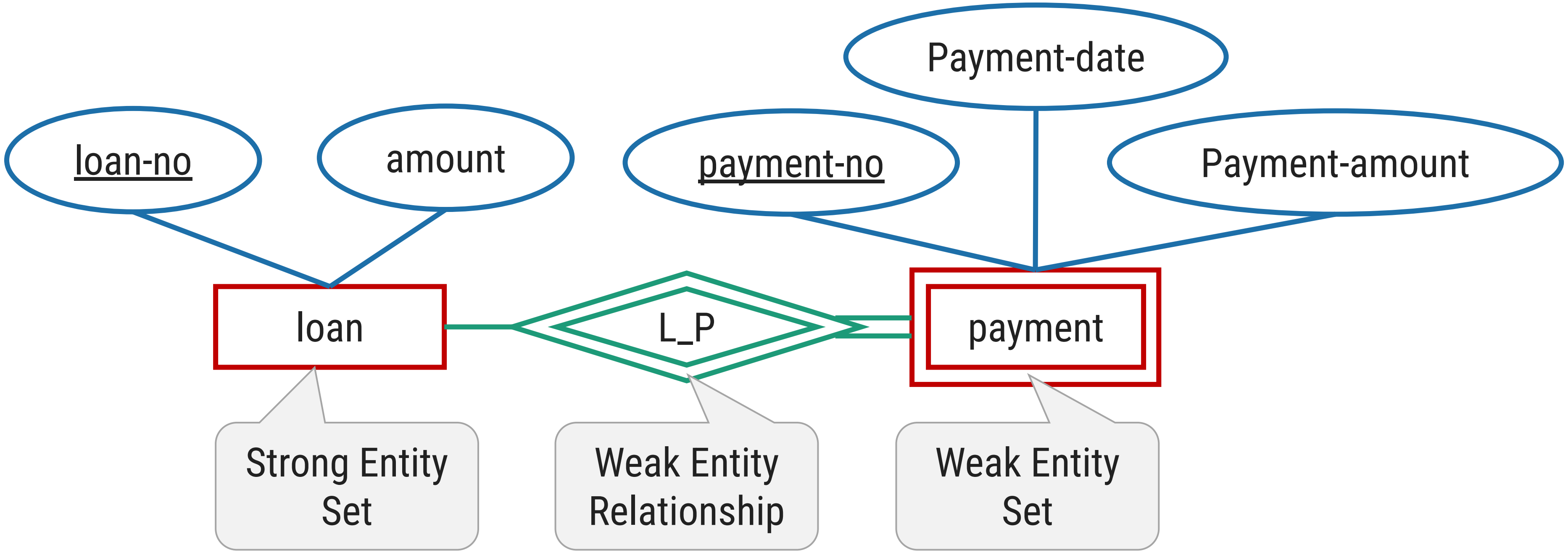
- every entity in the entity set participates in at least one relationship in the relationship set.
- indicated by double line



Each customer has maximum one loan

Weak Entity Set

- An **entity set that does not have a primary key** is called weak entity set.



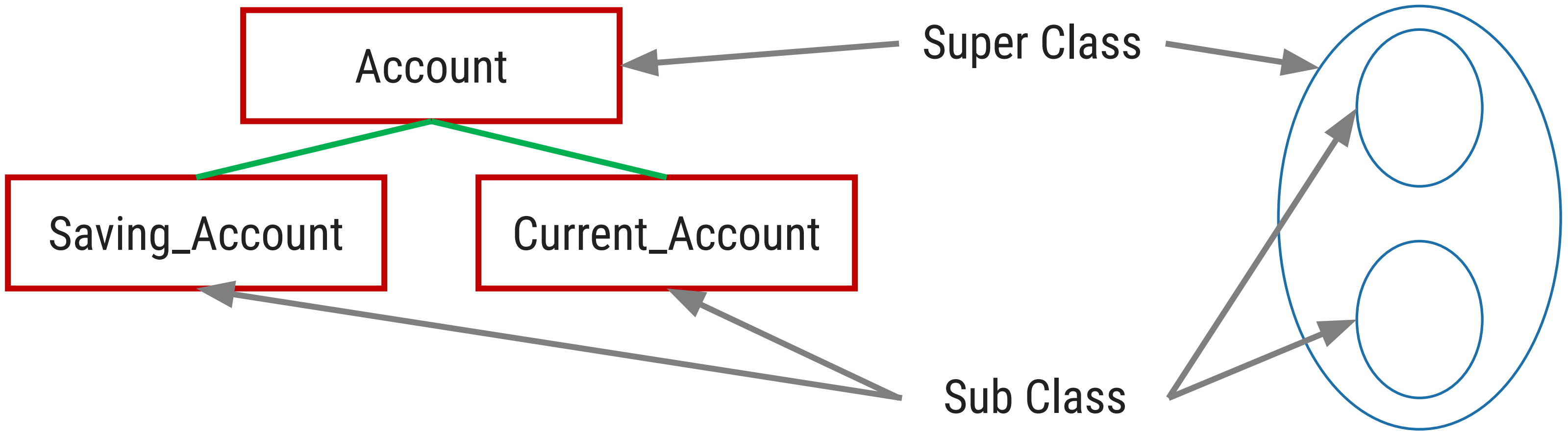
- Weak entity set is indicated by double rectangle.
- Weak entity relationship set is indicated by double diamond.

Weak Entity Set

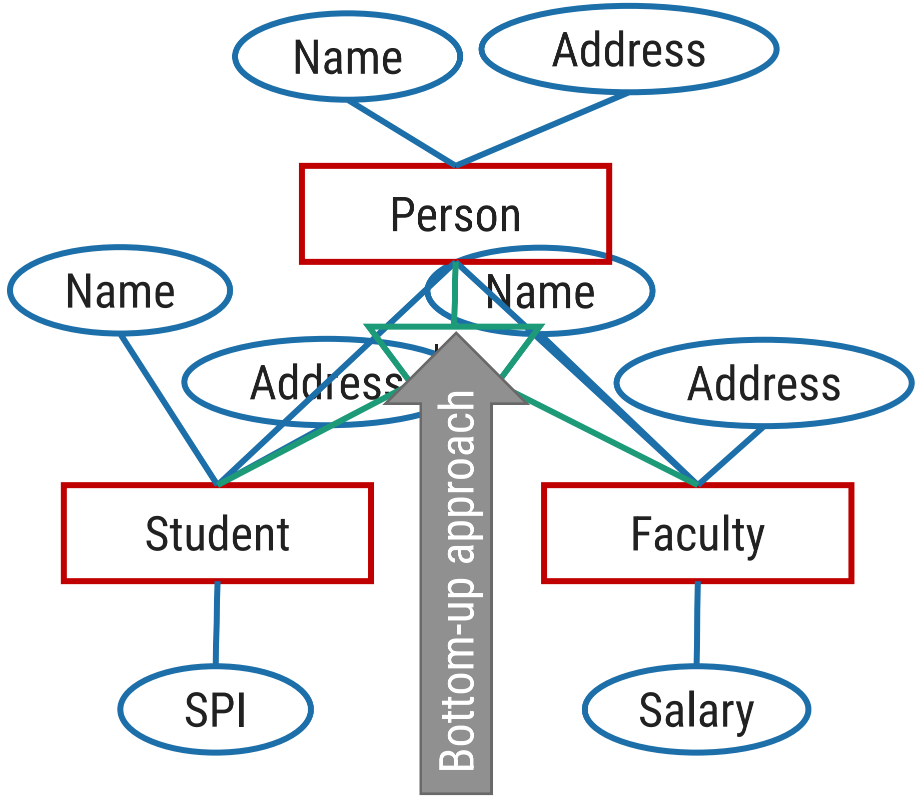
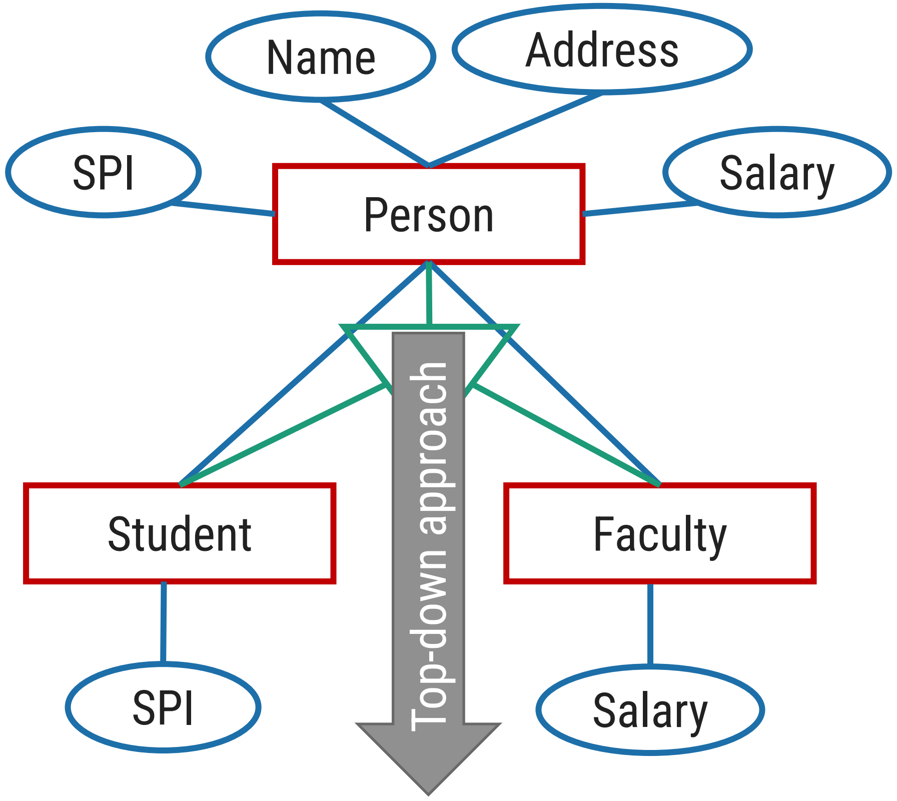
- The **existence of a weak entity set** depends on the **existence of a strong entity set**.
- The **discriminator (partial key)** of a weak entity set is the set of **attributes that distinguishes all the entities** of a weak entity set.
- The **primary key** of a weak entity set is created by **combining the primary key of the strong entity set** on which the weak entity set is existence dependent and the **weak entity set's discriminator**.
- We underline the discriminator attribute of a weak entity set with a **dashed line**.
- Payment entity has payment-no which is discriminator.
- Loan entity has loan-no as primary key.
- So primary key for payment is **(loan-no, payment-no).**

Superclass v/s Subclass

Super Class	Sub Class
A superclass is an entity from which **another entities can be derived**.	A subclass is an entity that is **derived from another entity**.
E.g, an entity account has two subsets saving_account and current_account So an **account is superclass**.	E.g, saving_account and current_account entities are derived from entity account. So **saving_account and current_account are subclass**.



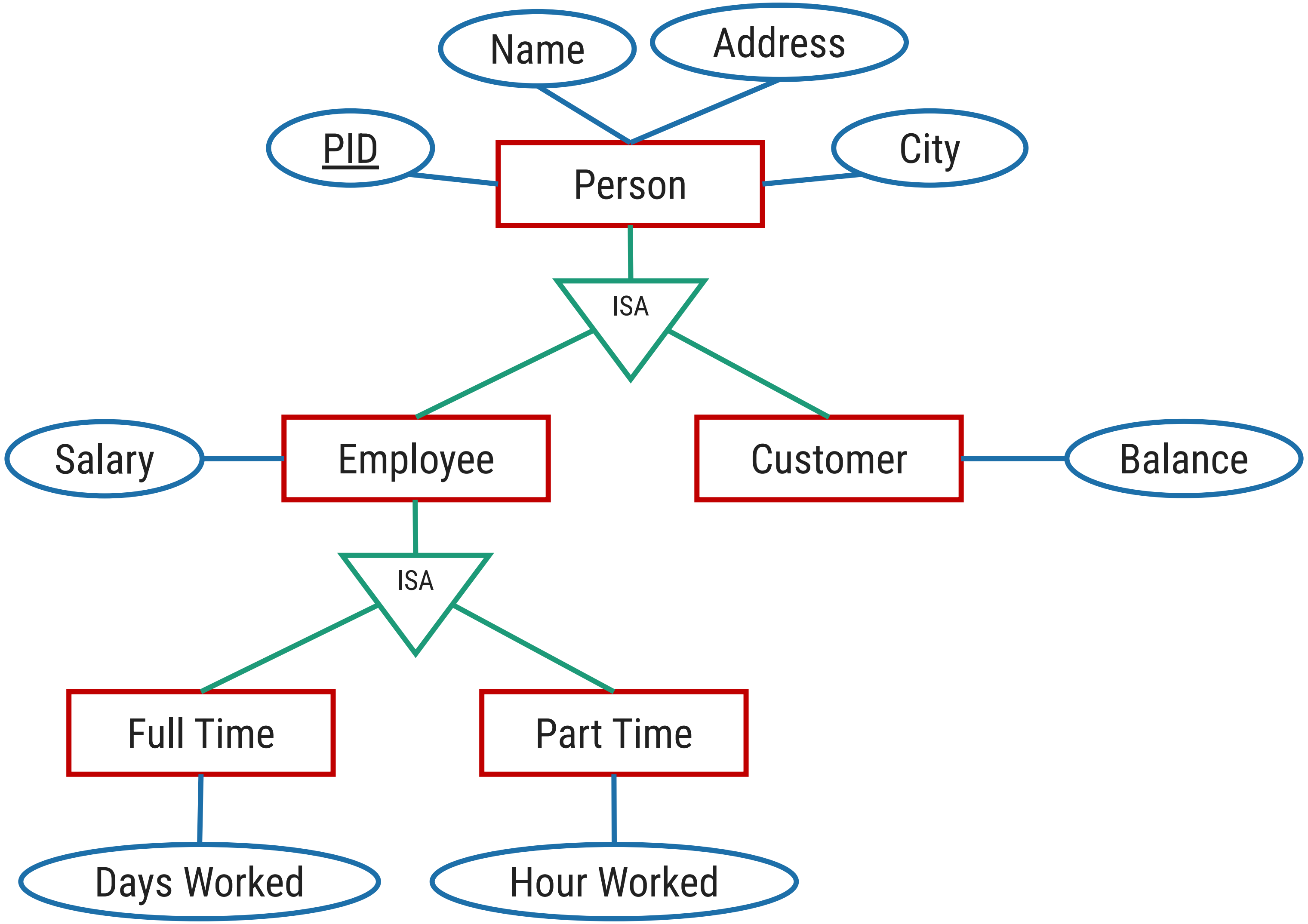
Generalization v/s Specialization

Generalization	**Specialization**
It **extracts the common features** of **multiple entities** to **form a new entity**.	It **splits an entity to form multiple new entities** that **inherit some feature of the splitting entity**.
 Name Address Person Name Name Address Address Student Faculty SPI Salary Bottom-up approach	 Name Address SPI Person Salary Student Faculty SPI Salary Top-down approach

Generalization v/s Specialization

Generalization	Specialization
The process of **creation of group from various entities** is called generalization.	The process of **creation of sub-groups within an entity** is called specialization.
It is **Bottom-up** approach.	It is **Top-down** approach.
The process of taking the **union of two or more lower level entity** sets to produce a higher level entity set.	The process of taking a **sub set of higher level entity set** to form a lower level entity set.
It starts from the number of entity sets and creates high level entity set using some common features.	It starts from a single entity set and creates different low level entity sets using some different features.

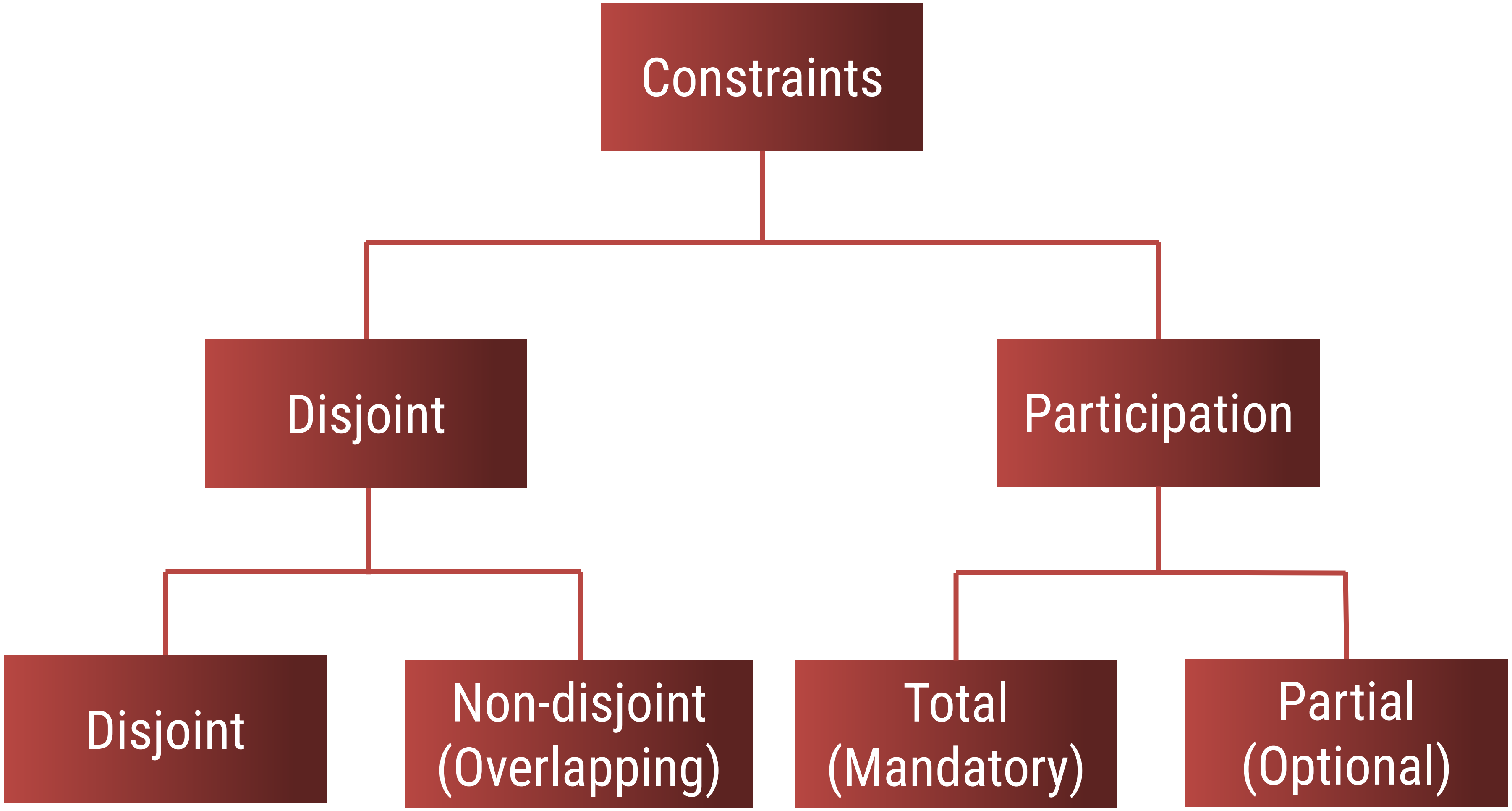
Generalization & Specialization example



Exercise

- Give the examples of Generalization/Specialization in the following E-R diagram:
  - Hospital Management System.
  - College Management System.
  - Bank Management System.
  - Insurance Company.

Constraints on Specialization and Generalization

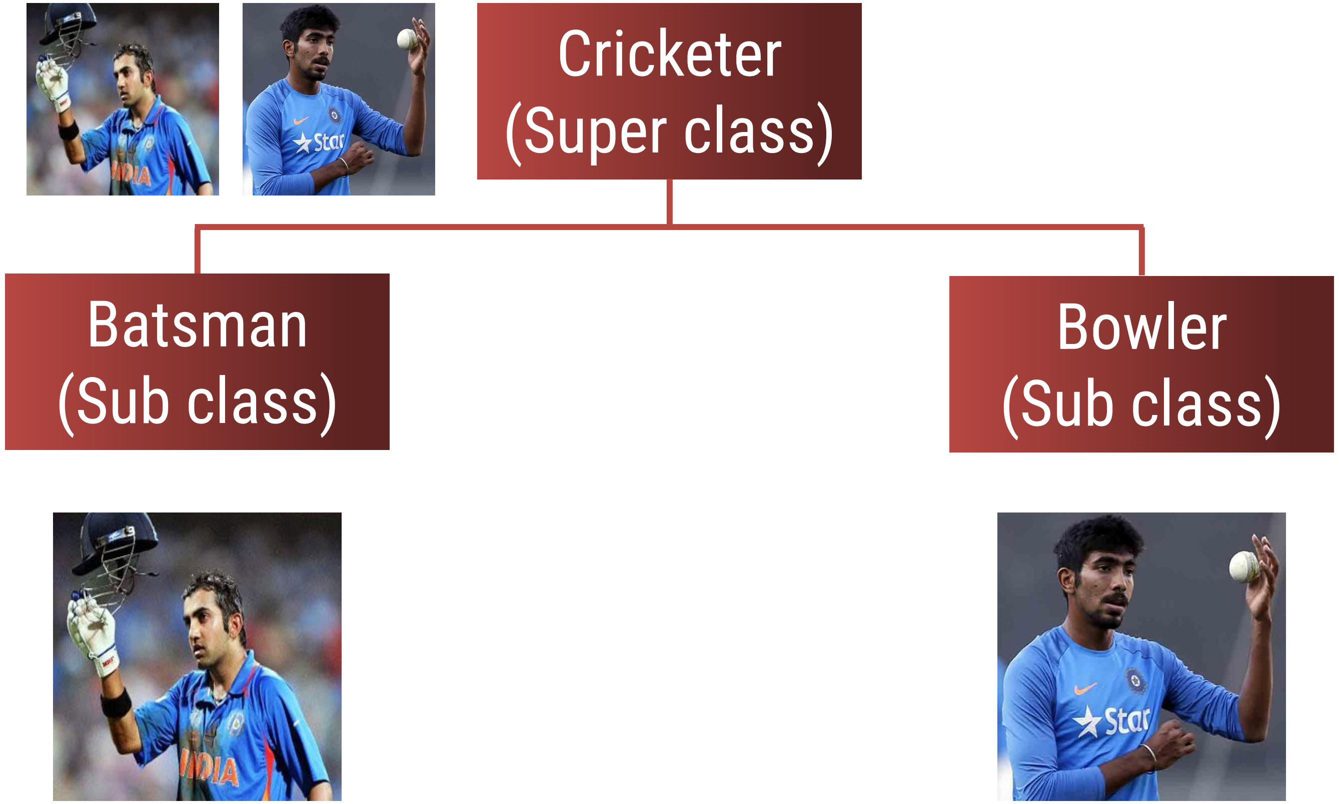


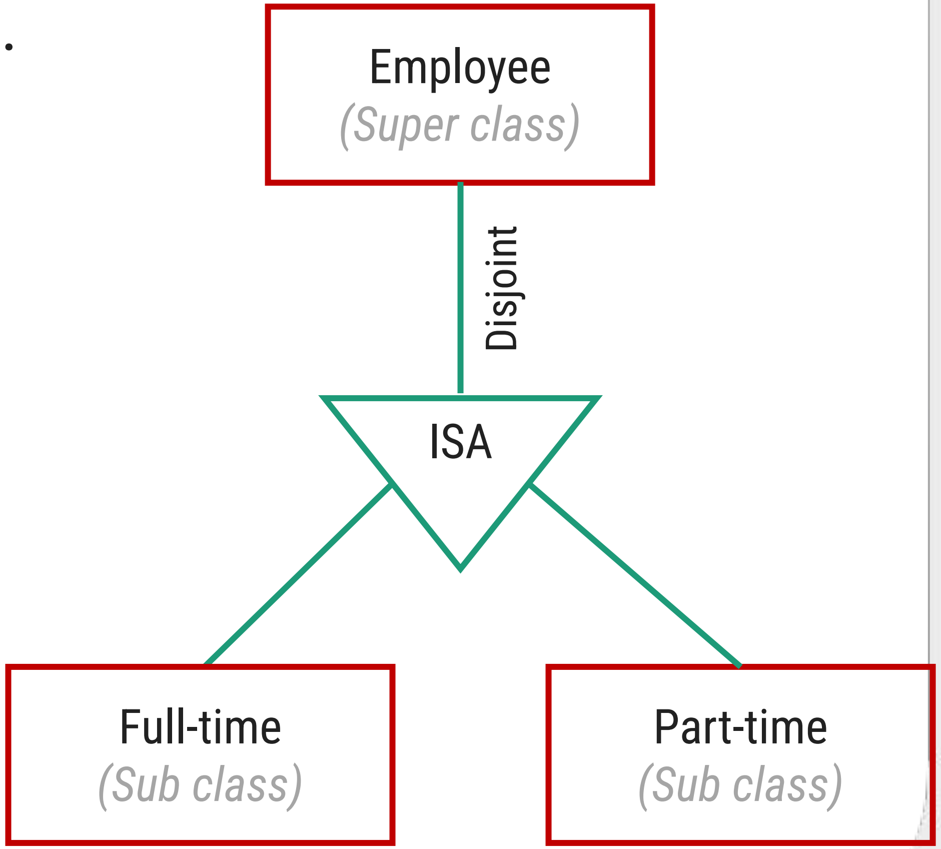
Disjoint Constraint

- It describes **relationship between members of the superclass and subclass** and indicates whether member of a superclass can be a member of one, or more than one subclass.
- Types of disjoint constraints
  - Disjoint Constraint
  - Non-disjoint (Overlapping) Constraint

Disjoint Constraint

- It specifies that the **entity of a super class can belong to only one lower-level entity set** (sub class).
- Specified by ‘**d**’ or by writing **disjoint** near to the ISA triangle.

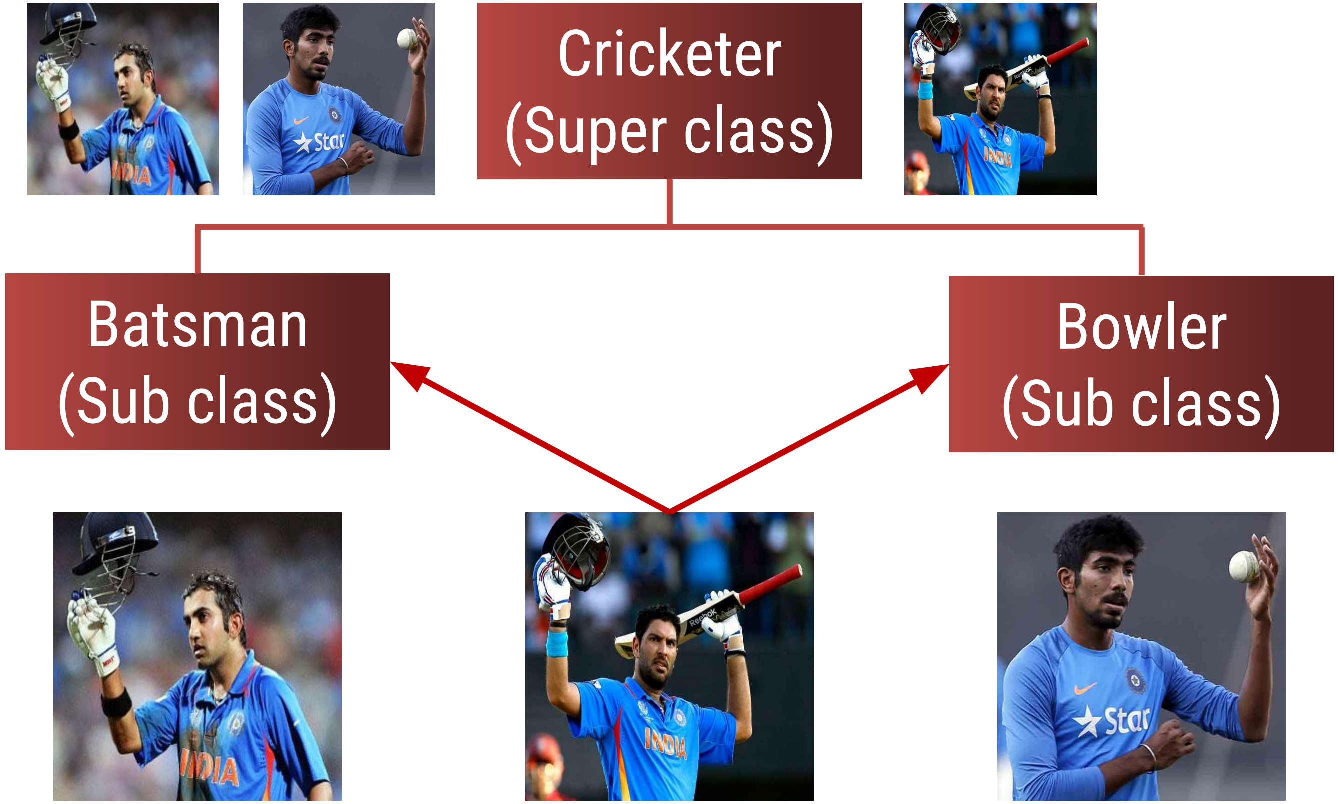




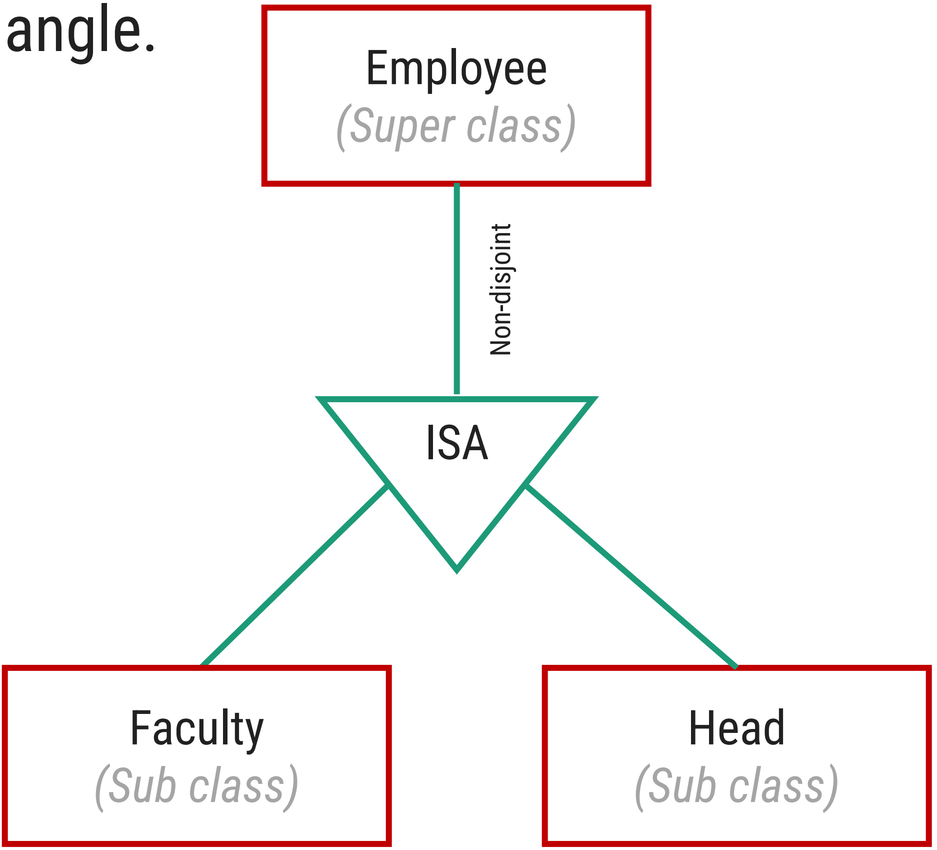
All the players are associated with only one sub class either (Batsman or Bowler).

Non-disjoint (Overlapping) Constraint

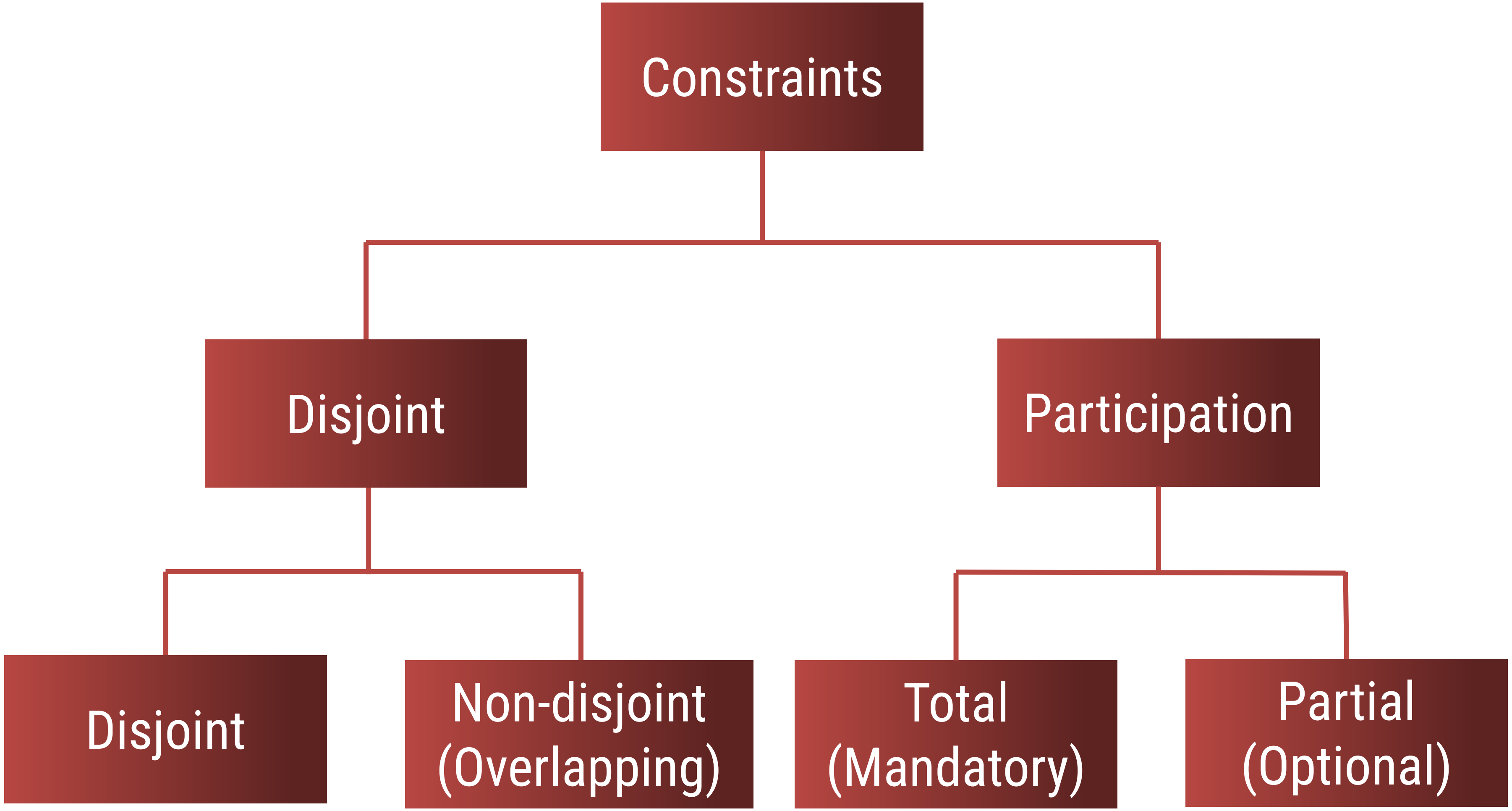
- It specifies that an **entity of a super class can belong to more than one lower-level entity** set (sub class).
- Specified by ‘**o**’ or by writing **overlapping** near to the ISA triangle.



One player (Yuvraj singh) is associated with more than one sub class.



Constraints on Specialization and Generalization

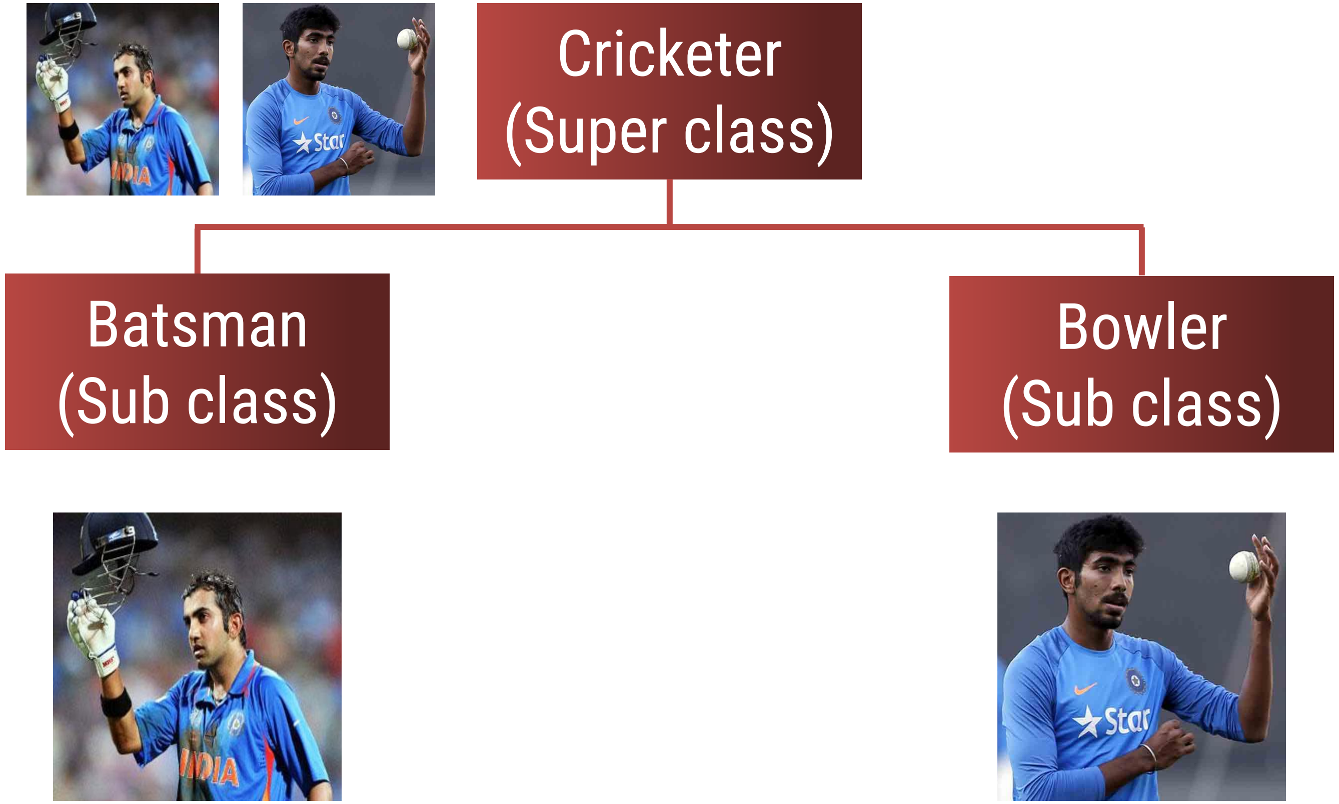


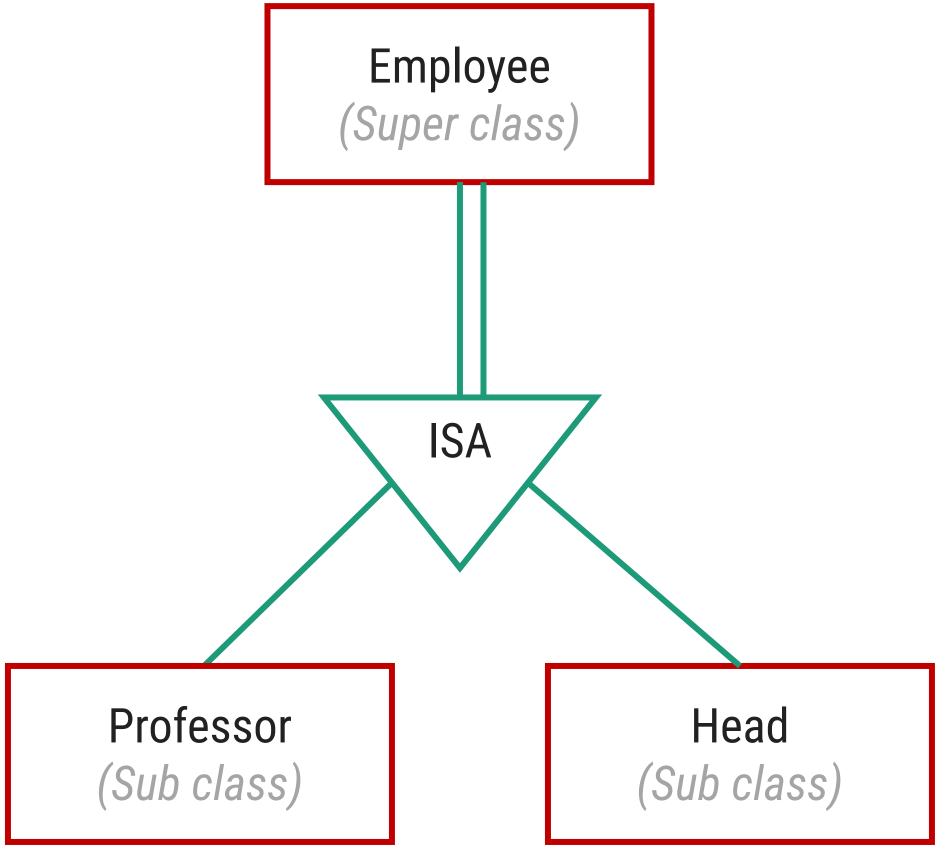
Participation (Completeness) Constraint

- It determines **whether every member of super class must participate as a member of subclass or not**.
- Types of participation (Completeness) Constraint
  - Total (Mandatory) participation
  - Partial (Optional) participation

Total (Mandatory) Participation

- Total participation specifies that **every entity in the superclass must be a member of some subclass** in the specialization.
- Specified by a **double line** in E-R diagram.

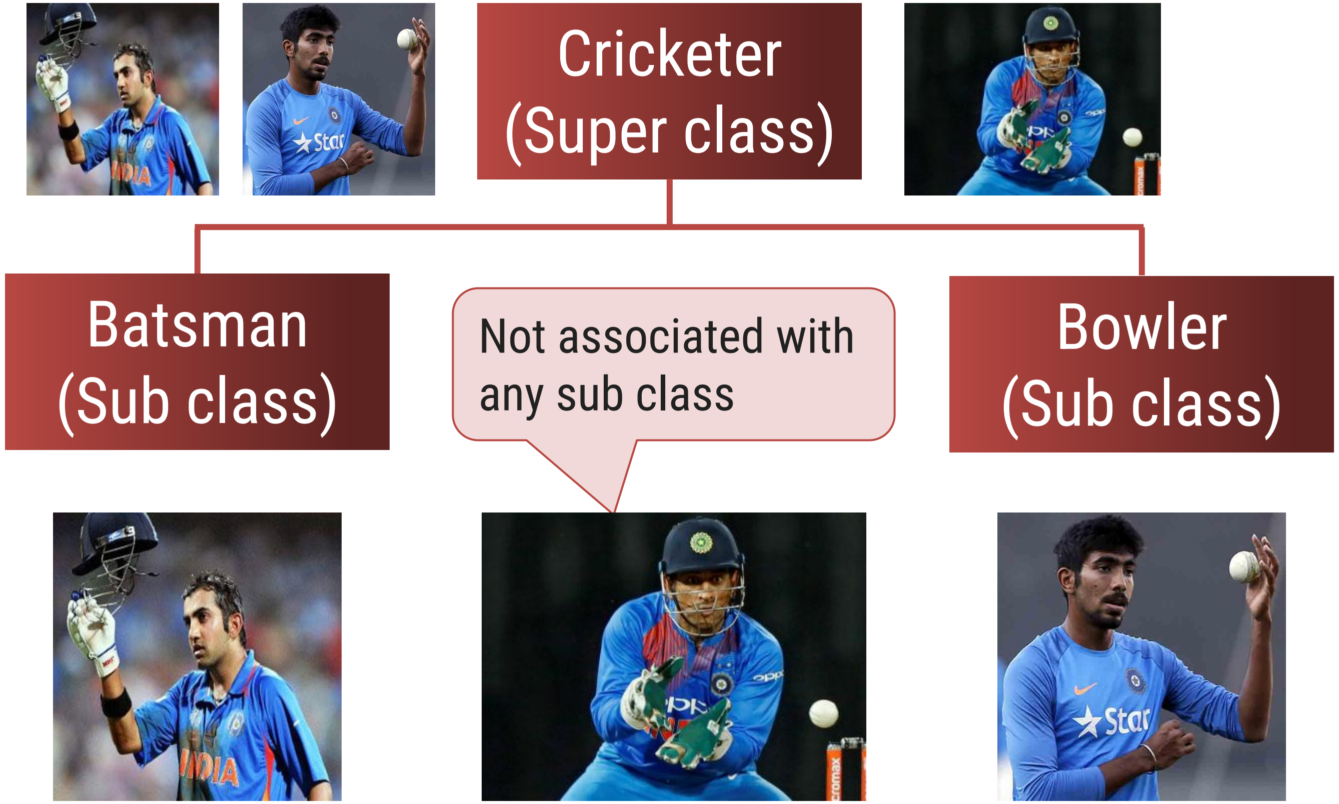




All the players are associated with minimum one sub class either (Batsman or Bowler).

Partial (Optional) Participation

- Partial participation specifies that **every entity in the super class does not belong to any of the subclass** of specialization.
- Specified by a **single line** in E-R diagram.

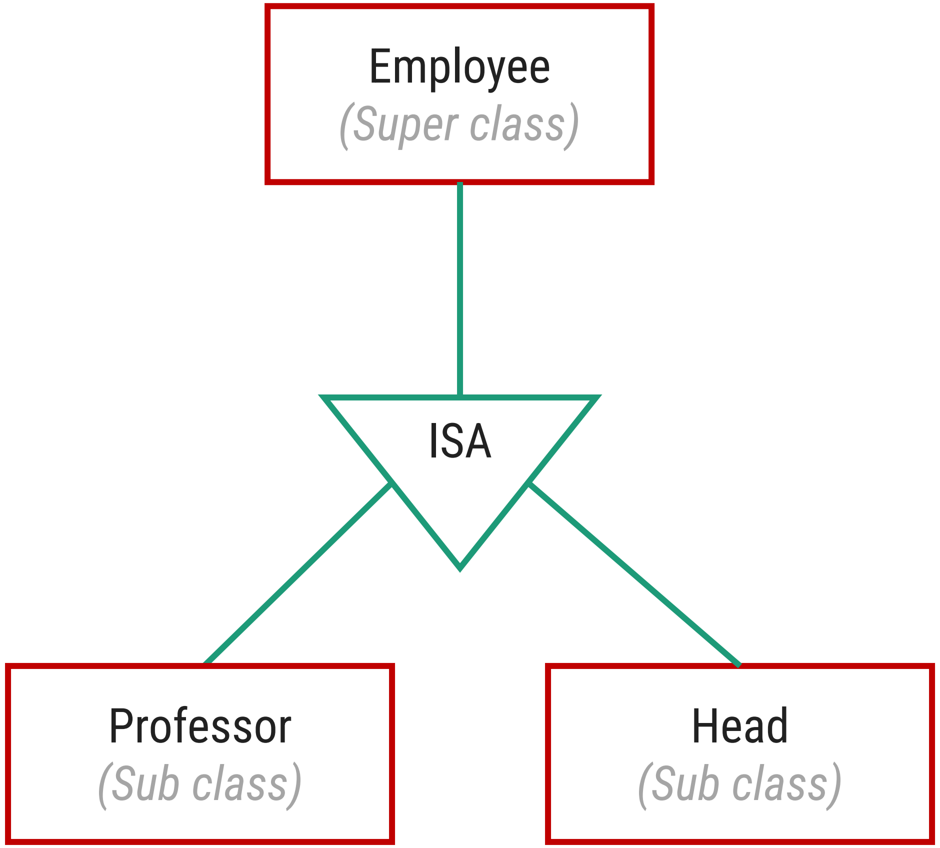


Cricketer
(Super class)

Batsman
(Sub class)

Not associated with any sub class

Bowler
(Sub class)



Employee
(Super class)

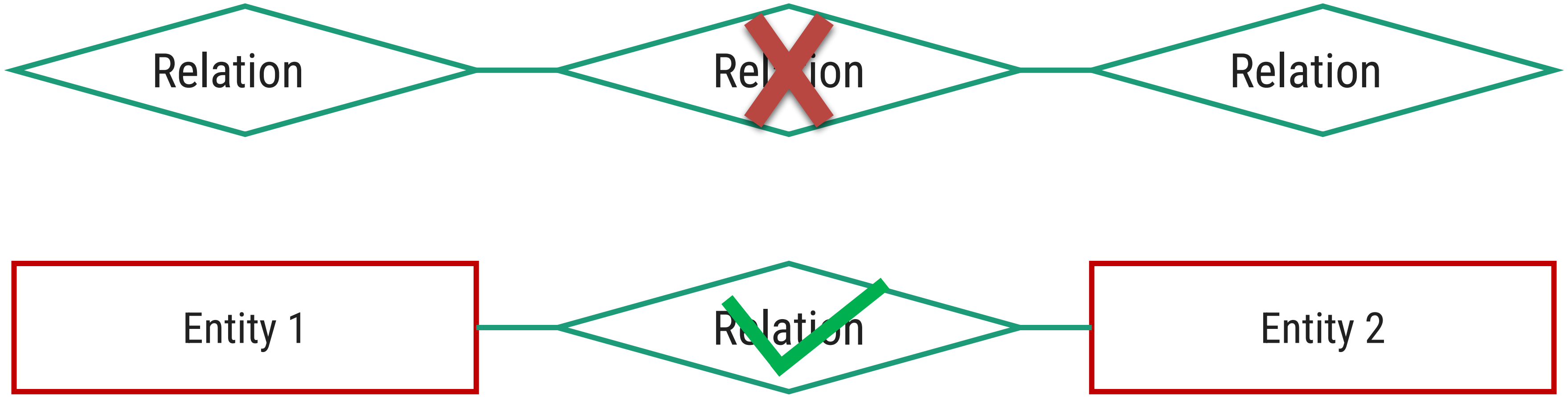
ISA

Professor
(Sub class)

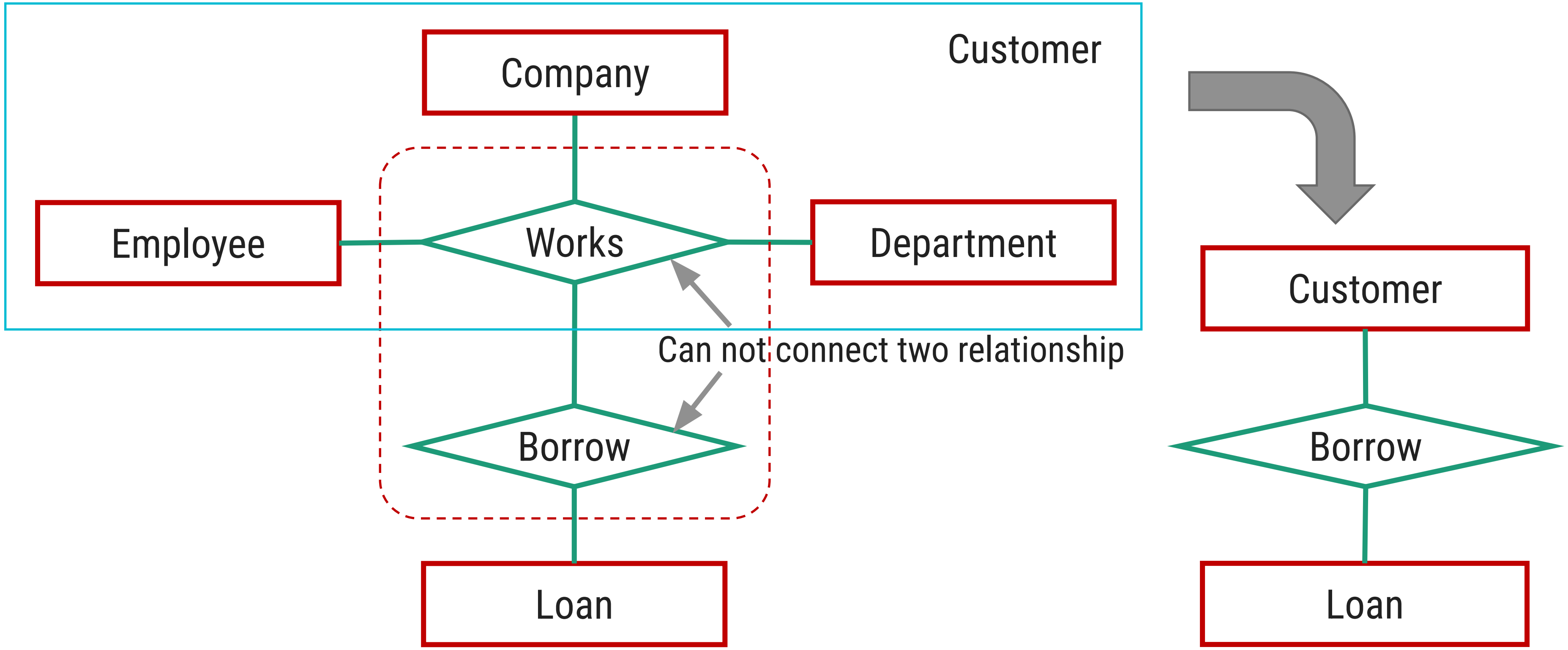
Head
(Sub class)

Limitation of E-R diagram

- In E-R model we **cannot express relationships between two relationships**.

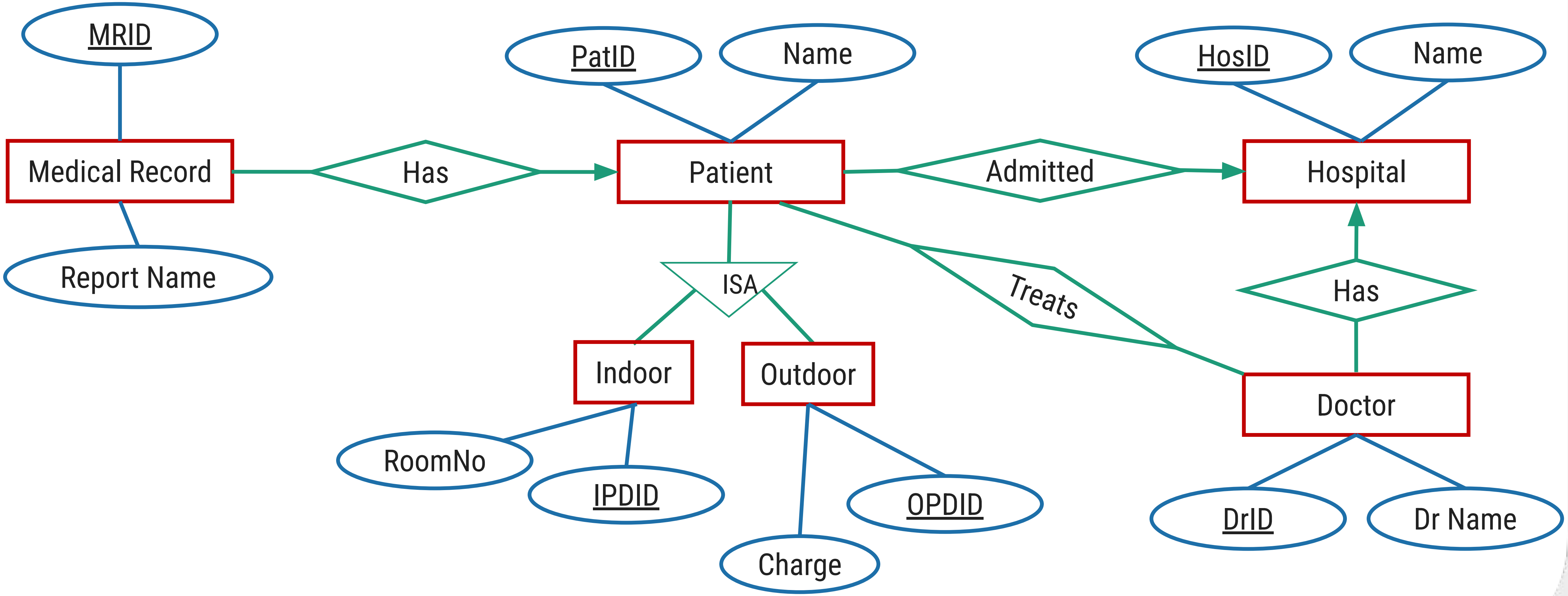


Limitation of E-R diagram



Process of creating an entity by combining various components of E-R diagram is called aggregation.

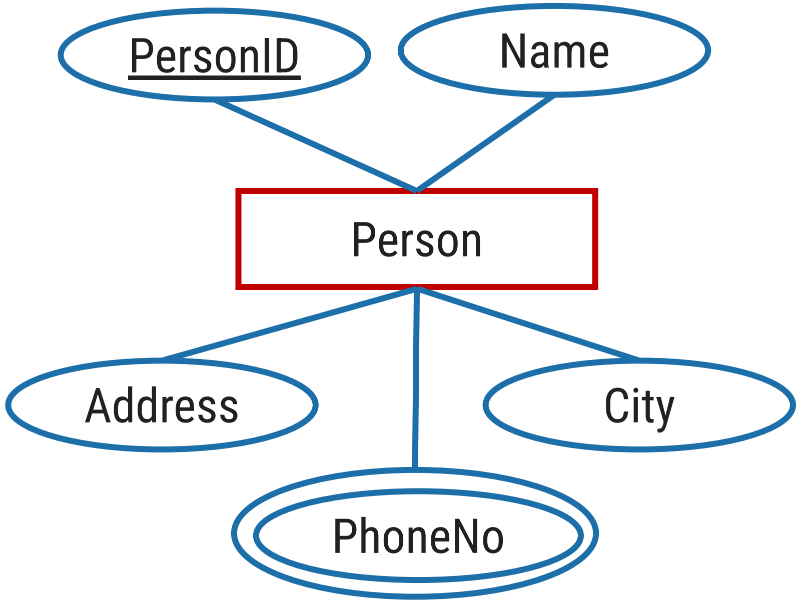
E-R diagram of Hospital Management System



Reduce the E-R diagram to database schema

Step 1: Reduce **Entities** and **Simple Attributes**:

- An **entity** of an ER diagram is **turned into** a **table**.
- Each **attribute** (except multi-valued attribute) **turns into** a **column** (attribute) in the table.
- **Table name** can be same as **entity name**.
- **Key attribute** of the entity is the **primary key** of the table which is usually underlined.
- It is highly recommended that every table should start with its primary key attribute conventionally named as TablenameID.

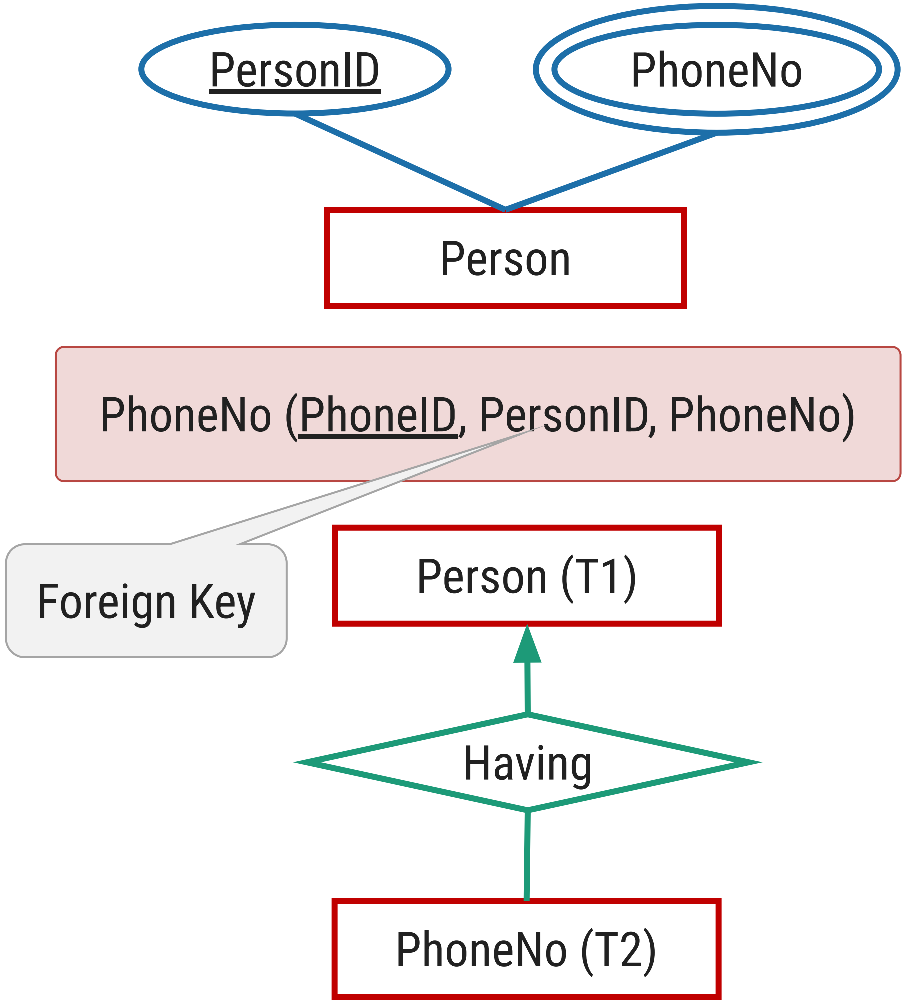


Person (PersonID, Name, Address, City)

Reduce the E-R diagram to database schema

Step 2: Reduce **Multi-valued Attributes**:

- **Multi-value attribute** is turned into a **new table**.
- **Add** the **primary key** column into **multi-value attribute's table**.
- Add the **primary key column** of the **parent entity's table** as a **foreign key** within the **new (multi-value attribute's) table**.
- Then make a **1:N relationship** between the Person table and PhoneNo table.



PhoneNo (PhoneID, PersonID, PhoneNo)

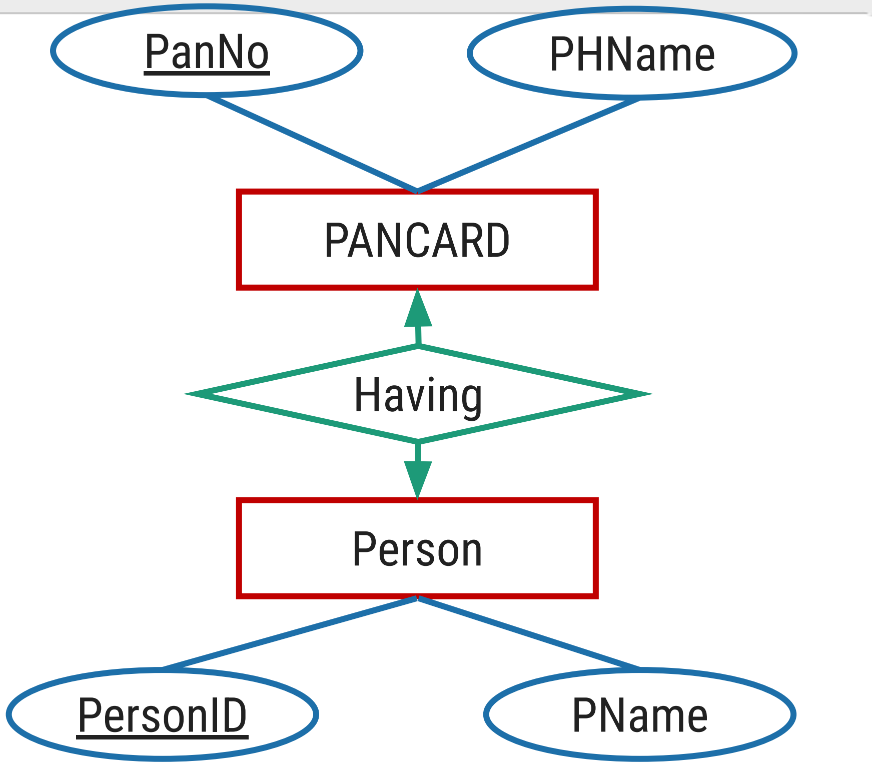
Reduce the E-R diagram to database schema

Step 3: Reduce **1:1 Mapping Cardinality**:

- Convert **both entities** in to **table** with proper attribute.
- Place the **primary key** of any **one table** in to the **another table** as a **foreign key**.
- Place the primary key of the PANCARD table PanNo in the table Persons as Foreign key.

OR

- Place the primary key of the Person table PersonID in the table PANCARD as Foreign key.



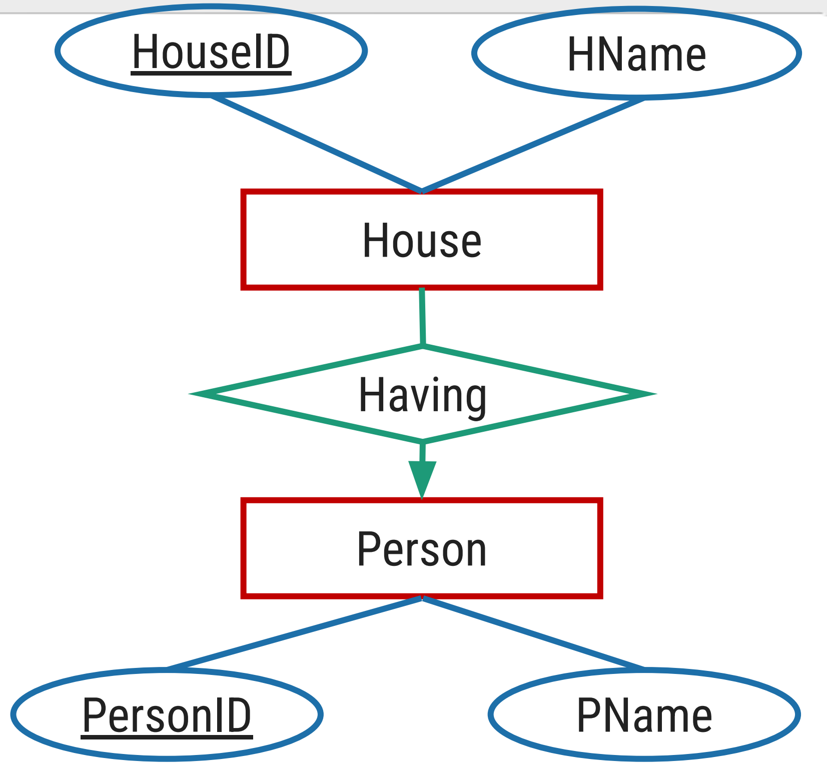
Person (PersonID, PName)
PANCARD (PanNo, PHName, PersonID)

PANCARD (PanNo, PHName)
Person (PersonID, Pname, PanNo)

Reduce the E-R diagram to database schema

Step 4: Reduce **1:N Mapping Cardinality**:

- Convert **both entities** in to **table** with proper attribute.
- Place the **primary key** of **table having 1 mapping** in to the another **table having many cardinality as a Foreign key**.
- Place the primary key of the Person table PersonID in the table House as Foreign key.

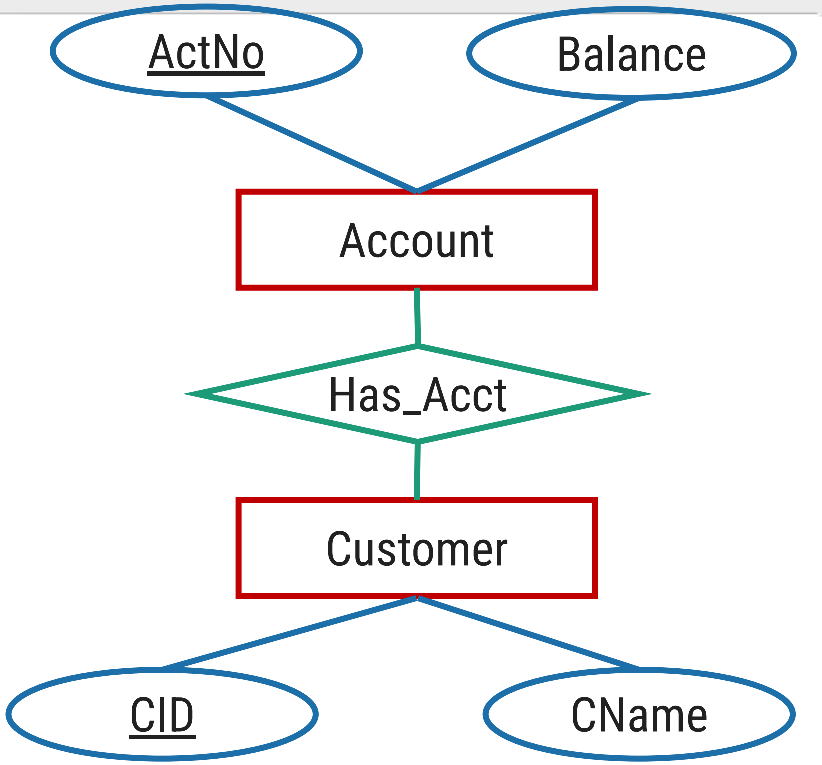


Person (PersonID, PName)
House (HouseID, Hname, PersonID)

Reduce the E-R diagram to database schema

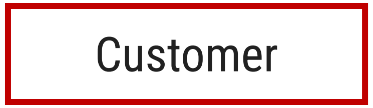
Step 5: Reduce **N:N Mapping Cardinality**:

- Convert both **entities** in to **table** with proper attribute.
- Create a **separate table for relationship**.
- Place the **primary key of both entities table** into the **relationship's table as foreign key**.
- Place the primary key of the Customer table CID and Account table Ano in the table Has_Acct as Foreign key.

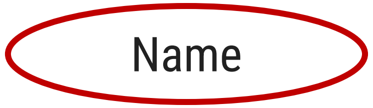


Customer (CID, CName)
Account (ActNo, Balance)
Has_Acct (HasAcctID, CID, ActNo)

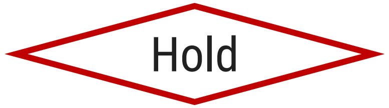
Summery of Symbols used in E-R diagram



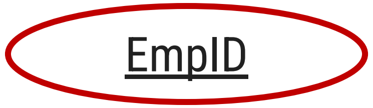
Entity



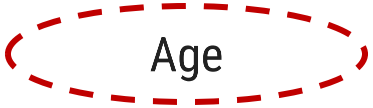
Attribute



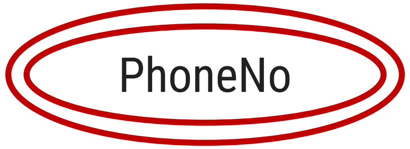
Relationship



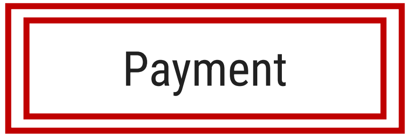
Primary Key Attribute



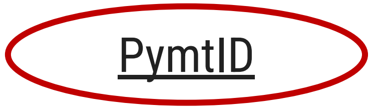
Derived Attribute



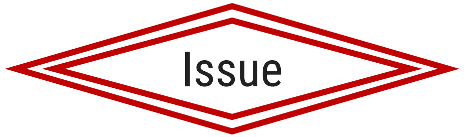
Multi Valued Attribute



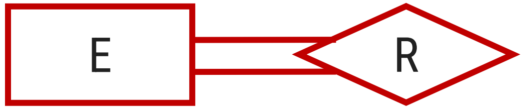
Weak Entity



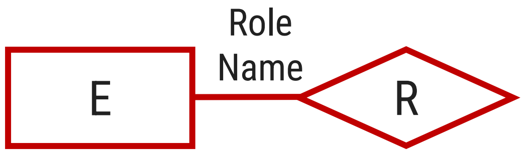
Discriminating Attribute



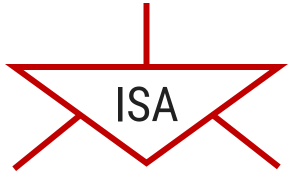
Weak Entity Relationship



Total Participation

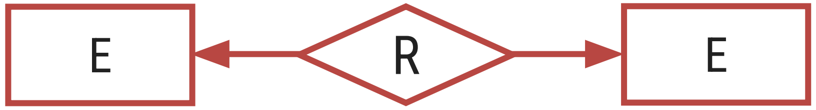


Role Indicator

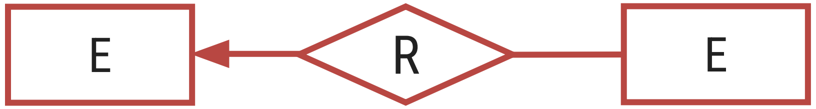


Specialization/ Generalization

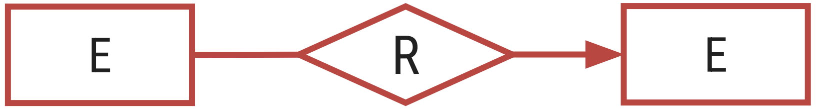
Summery of Symbols used in E-R diagram



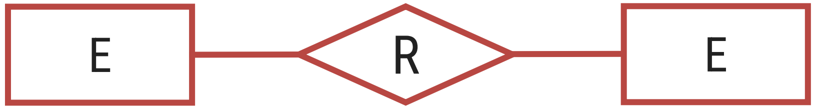
One to One



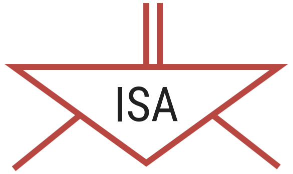
One to Many



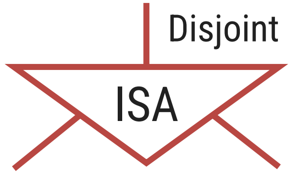
Many to One



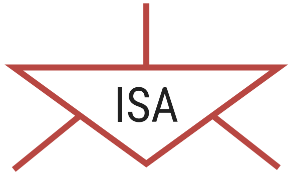
Many to Many



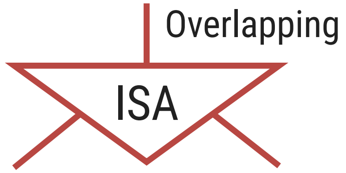
Total Specialization/ Generalization



Disjoint Specialization/ Generalization



Partial Specialization/ Generalization



Overlapping Specialization/ Generalization

Integrity Constraints

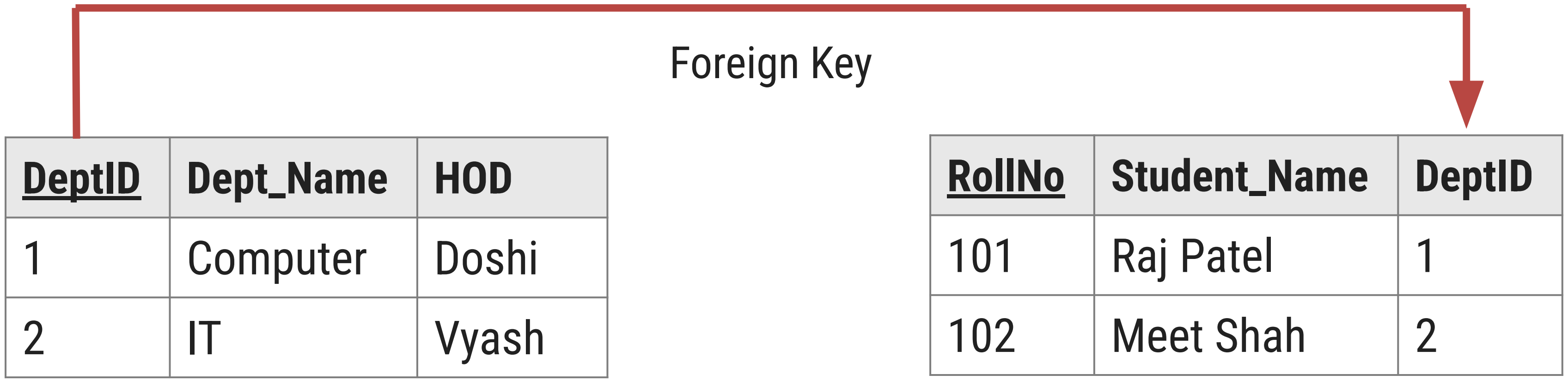
- Integrity constraints are a **set of rules**. It is used to **maintain the quality** of information.
- Integrity constraints ensure that the data insertion, updating, and other processes have to be performed in such a way that data integrity is not affected.
- Thus, integrity constraint is used to **guard against accidental damage** to the database.
- Various Integrity Constraints are:
  - Check
  - Not null
  - Unique
  - Primary key
  - Foreign key

Integrity Constraints

- Check
  - This constraint defines a business rule on a column. All the rows in that column must satisfy this rule.
  - Limits the data values of variables to a **specific set, range, or list of values**.
  - The constraint can be applied for a single column or a group of columns.
  - E.g. value of SPI should be between 0 to 10.
- Not null
  - This constraint ensures all rows in the table contain a definite value for the column which is specified as not null. Which means a **null value** is not allowed.
  - E.g. name column should have some value.
- Unique
  - This constraint ensures that a column or a group of columns in each row have a **distinct (unique)** value.
  - A column(s) can have a null value but the values cannot be duplicated.
  - E.g. enrollmentno column should have unique value.

Integrity Constraints

- Primary key
  - This constraint defines a column or combination of columns which uniquely identifies each row in the table.
  - Primary key = **Unique key + Not null**
  - E.g. enrollmentno column should have unique value as well as can't be null.
- Foreign key (referential integrity constraint)
  - A referential integrity constraint (foreign key) is specified between two tables.
  - In the referential integrity constraints, if a foreign key column in table 1 refers to the primary key column of table 2, then every value of the foreign key column in table 1 must be null or be available in primary key column of table 2.



Foreign Key

DeptID	**Dept_Name**	**HOD**
1	Computer	Doshi
2	IT	Vyash

RollNo	**Student_Name**	**DeptID**
101	Raj Patel	1
102	Meet Shah	2

Questions asked in Exam

1. Write a note on mapping cardinality in E-R diagram.
2. Explain the difference between a weak and a strong entity set.
3. Explain the difference between generalization and specialization. **OR** Explain specialization and generalization concept in E-R diagram with suitable example.
4. Write a note on constraints on specialization and generalization.
5. Explain aggregation in E-R diagram with example.
6. What do you mean by integrity constraints? Discuss various integrity constraints.

Questions asked in Exam [E-R diagrams]

7. Draw E-R diagram for Bank Management System.
8. Define E-R diagram. Draw an E-R diagram for Library Management System. Assume relevant entities and attributes for the given system.
9. 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.
10. Design a generalization–specialization hierarchy for a motor-vehicle sales company. The company sells motorcycles, passenger cars, vans, and buses. Justify your placement of attributes at each level of the hierarchy. Explain why they should not be placed at a higher or lower level.

Questions asked in Exam [E-R diagrams and Database]

11. Design a database for an airline. 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. Your design should include an E-R diagram, a set of relational schemas, and a list of constraints, including primary-key and foreign-key constraints.
12. Design a database for a hospital with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examinations conducted. Your design should include an E-R diagram, a set of relational schemas, and a list of constraints, including primary-key and foreign-key constraints.

Thank You