**Chapter 3: DATA MODELS**

**Topic – 1: Models**

**Data Models**

* Represents logical structure of database.
* Are blueprints for DBMS abstraction.

**DBMS Models**

* Entity relationship model (ER)
* Relational model
* Network model
* Object oriented model

**ER Model**

* ER diagram is graphical representation of database.
* Creates – Entity set, relationship set, general attributes & constraints.

**Topic – 2: Relationship Diagram**

**Entity**

* Real world object/place/person.
* Represented by rectangle.

**Attributes**

* Properties of an entity.
* Represented by oval.

**Relationship**

* Represented by diamond.
* Is drawn between two entities.

**Topic – 3: Types Of Attributes**

**Introduction**

* Simple and composite
* Single-valued and multi-valued
* Stored attribute and derived attribute
* Complex attribute
* Key attribute

**Simple & Composite Attributes**

* **Simple:** Can’t be divided into subparts.
* **Composite:** Can be divided into subparts.

**Single & Multi-Valued Attributes**

* **Single:** Have single value.
* **Multi:** Represented by double oval.

**Stored & Derived Attributes**

* **Stored:** Have to be entered manually.
* **Derived:** Can be calculated by pre-existing attributes, represented by dashed oval.

**Complex Attributes**

* Derived by joining composite and multi-valued attribute.

**Key Attribute**

* Attribute which is unique to each entity.

**Descriptive Attributes**

* Attribute of a relation.

**Topic – 4: Relationship Sets**

**Recursive Relationship Set**

* An entity participating in multiple relations.

**Degree Of Relationship Set**

It is the number of entity sets in a relationship.

* Unary relationship (one entity relation, **ex:-** Marriage (two line representation))
* Binary relationship (two entity relation)
* N-ary/ternary relationship (three (ternary) or more (N-ary) entity relation)

**Cardinality Constraints**

* Also known as **mapping constraints**.
* Number of times an entity of another set participates in a relationship.
* Helps in defining binary relations.
* Binary relationships mapping are:-
  + **One to one** (an entity in A is related to one entity in B)
  + **One to many** (an entity in A is related to multiple entities in B)
  + **Many to one** (opposite of previous)
  + **Many to many** (cross connections)

**Weak Entity Set**

* Doesn’t have a primary key.
* Also called dependant entity.
* Represented by double rectangle.

**Weak Entity Relationship**

* Involves at least one weak entity.
* Represented by double diamond.

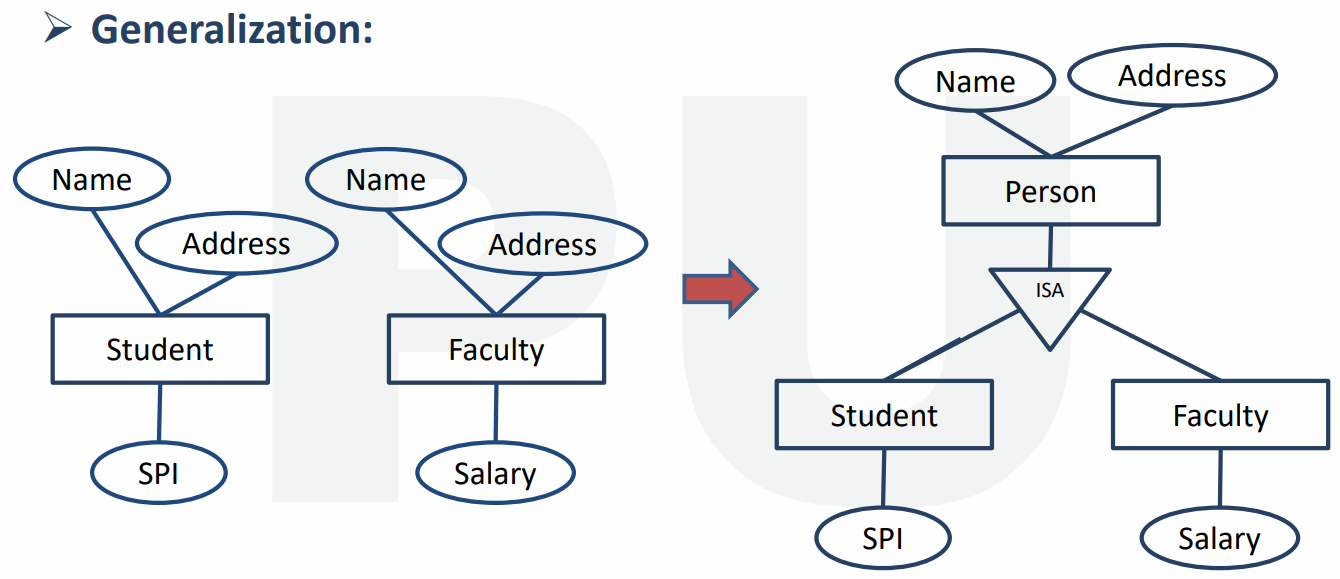
**Super Class & Sub Class**

* Other entities can be derived from superclass.

**Topic – 5: Constraints & Features**

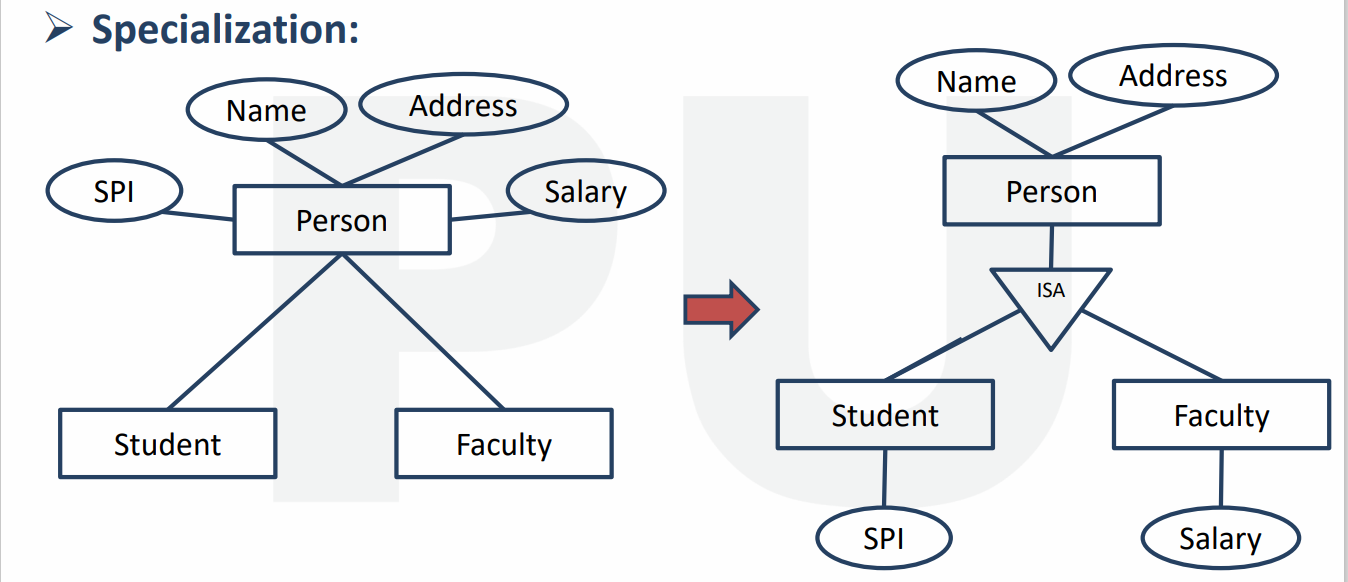
**Generalization**

* Selects common attributes from two or more entities, and makes a separate entity out of it.
* Same triangle representation is involved as was in generalization.



**Specialization**

* Splits an entity, and makes multiple entities.
* The newly created entities inherit some attributes of the entity that split.



**Constraints On Generalization & Specialization**

* Disjoint constraint
  + Disjoint constraint
  + Non-disjoint constraint (overlapping)
* Participation constraint
  + Total participation (mandatory)
  + Partial participation (optional)

**Disjoint Constraint**

* Entity of a superclass can belong to one subclass entity set.
* Represented by writing **d** or **disjoint** between **superclass** and **ISA** triangle.

**Non-Disjoint Constraint**

* Same as previous, but could belong to multiple subclasses.

**Total Participation**

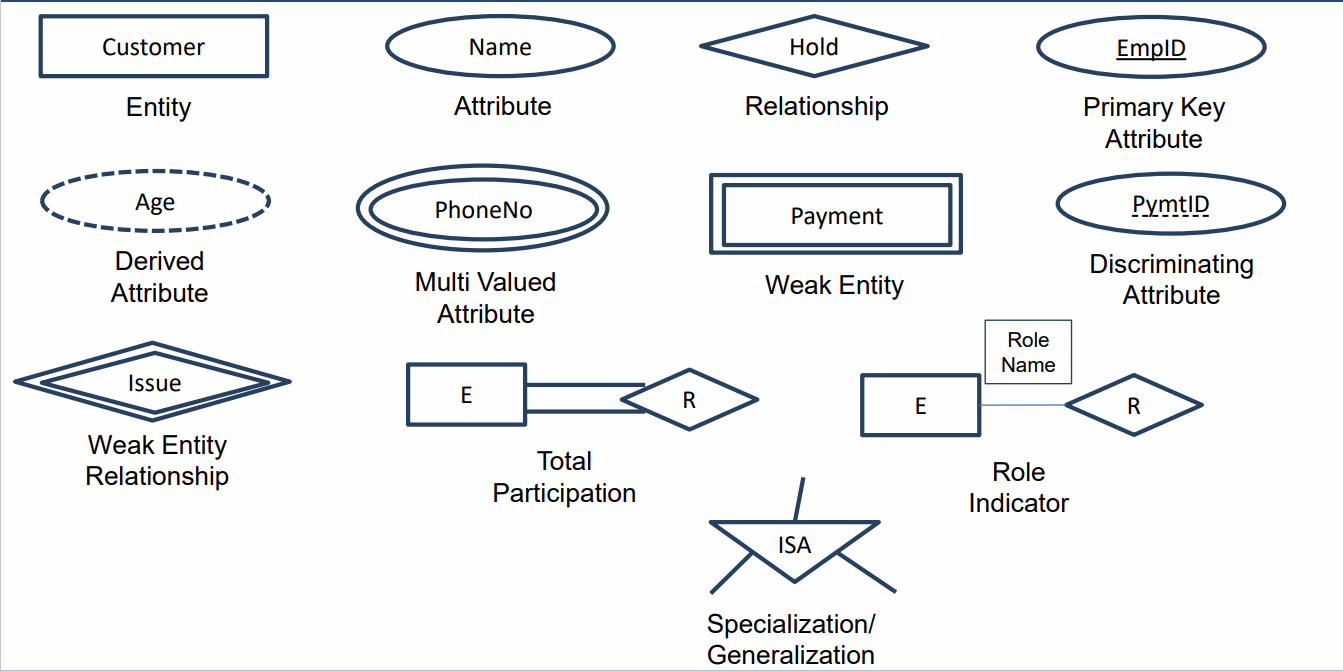
* Every entity in entity set participates in at least one relationship.
* Every superclass entity must be a subclass member.
* Represented by double line.

**Partial Participation**

* Some entities in entity set may **not** participate.
* Every superclass entity is **not** a **subclass** **member**.
* Represented by **single line**.

**Aggregation**

* Abstraction of ER models.
* Defining an entity briefly in diagramming.



**Models**

* Hierarchical (one to many)
* Network (many to many)

**Relational Models**

* Two-dimensional tables.
* Stores common attributes.
* Column = attribute
* **Relational instance:** Set of tuples.
* **Attribute domain:** Value constraint of attributes.

**Object Oriented Model**

* Object properties are called **attributes**.
* Object behaviour are called **methods**.

**Integrity Constraints**

* Makes sure that changes doesn’t affect the integrity of data.
* We discuss about keys in this.
* **These are:-** Check, not null, unique, primary key, foreign key.

**Check**

* Applied on columns.
* **Ex:-** CGPA column (between 0 and 10 only)

**Unique**

* Could be null but unique.

**Keys**

* **Primary key**
  + Unique + not null
* **Foreign key**
  + Also called referential integrity.
  + Must be null when initialized.
  + Not unique in other table.
* **Super key**
  + Identifies each tuple uniquely.
  + Can be one or more.
* **Candidate key**
  + Subset of super key.
  + Doesn’t repeat attributes.
  + Also called minimal super key.
  + Every table has at least one candidate key.
* **Alternate key**
  + Non-candidate key.