

CHAPTER 3  
EER

Step 1:

- Entity : an entity, which is a thing or object in the real world with an independent existence

physical  
car, house, employ

conceptual

company, job, university

Step 2:

- Each entity has attributes

↳ the particular properties that describe it.

Single-valued

multiple-valued

composite

derived

→ A person may have more than one degree

→ multivalued attributes

→ Age and Birth-date attribute of a person

↳ 'Age' → derived attribute

Step 3:

→ key → value can be used to identify each entity uniquely

e.g. Name attribute is key of company

because no two companies allowed to have

same name.



Tertiary Relationship

→ confusion

Step 4

→ Relationships.

Step 5

→ Cardinality ratio:-

maximum number of relationship instances  
that an entity can participate in.

Department: Employee → 1:N

1:1, 1:N, Nil and M:N

→ Step 6

Participation constraints

Partial

total / existence dependency

"we do not expect every employee must work for  
every employee to manage a department"

some department

entity via MANAGES

But not all

→ step 7 Attributes of a relationship type

## Step 8

- List Entity types:
  - ↳ Entity types that do not have key attributes of their own are called weak entity types.
- Always total participation constraint.

Two dependents of two distinct employees may have the same value for Name, Birth date, sex and relationship.  
↓  
↓ distinct  
particular employee entity  
to which each dependent is related
- A weak entity type normally has a partial key.
- Box and diamonds with a double line
- partial key underlined with dashed or dotted line
- Another notation for cardinality at least min and at most max

\* Confuse → address

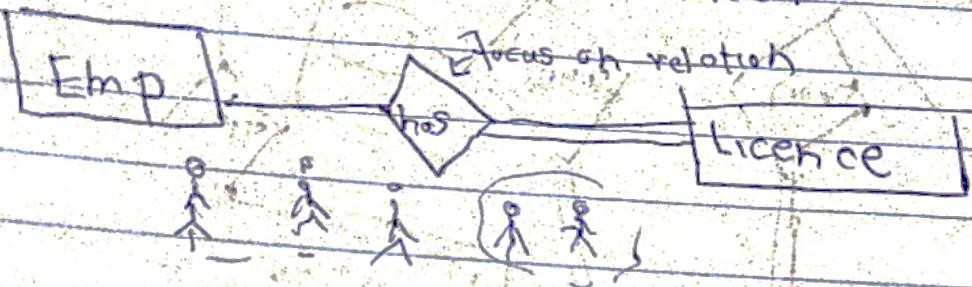
Note] How to make correct relationships.

→ How many entities name in a particular line. So make relationship between all the entities.

Total

and

Partial



If it is not necessary for employ has licence.

## CHAPTER 4 EER DIAGRAM

→ Specialization:

↳ specialization is process of defining a set of subclasses of an entity type; this entity type is called superclass of the specialization.

→ The set of subclasses that forms a specialization is defined on the basis of some distinguishing characteristics.

→ some real world entity

↓  
employ specialized in role of secretary

↓  
employ specialized in role of technician

method  
of pay

job type of  
each class

→ Generalization:

→ Common attributes generalized into superclass.

→ An arrow pointing to generalize superclass

is called generalization and an arrow pointing to specialized superclass is called specialization.

Constraints on specialization and generalization:

→ A specialization may also consist of a single subclass only, such as the Manager specialization. In such a case, we do not use the circle notation.

→ entity that will become members of each subclass by placing a condition on the value of some attribute of superclass. Such subclasses are called predicate-defined (or condition-defined) subclasses → e.g. Job-type.

→ disjointness constraint (d)  
→ subclass of specialization must be disjoint set.  
→ entity can be a member of at most one of subclasses of specialization  
→ d in circle stands for disjoint.

→ completeness or (totalness) constraint (t)  
which may be, partial or total.

↳ Total Specialization  
every entity in the superclass must be a member of at least one subclass

↳ Partial Specialization  
entity not belonging to any of the subclasses

→ We have four possible constraints on specialization.

Disjoint, total

Disjoint, partial

Overlapping, total

Overlapping, partial

→ In general, a superclass that was identified through the generalization process usually is total, because super class derived from Subclasses.

→ Specialization and Generalization Hierarchy:

Degree → Multivalued

Note: we draw a diagram for specialization must check that it disjointness or overlapping → engineer manager must exist in all three classes

→ A subclass with more than one superclass is called shared subclass, e.g. Student-Assistant.

→ Union type or category → owner may be bank, company and Person

→ A category has two or more superclasses

