Entity Relationship Model

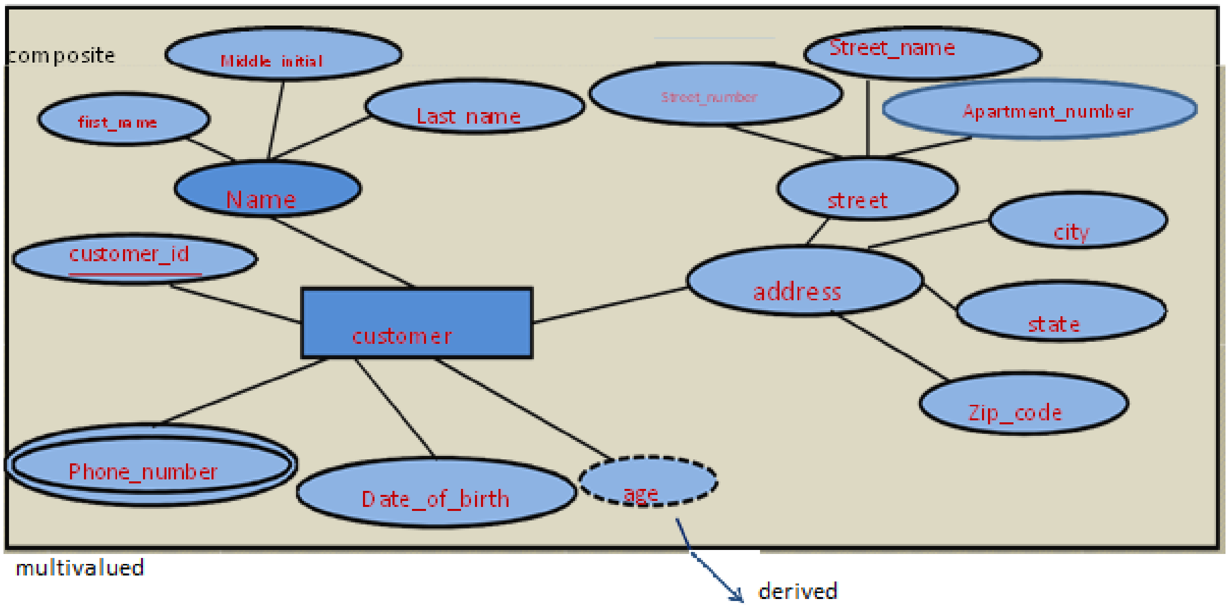
The ER model defines the conceptual view of a database. It works around real-world entities and the associations among them. It was developed to facilitate database design by allowing specification of an enterprise schema, which represents the overall logical structure of a database.
The ER model is very useful in mapping the meanings and interactions of real-world enterprise onto a conceptual schema. Because of this usefulness, many database-design tools draw on concepts from the ER model.

Steps used in ER modeling:

1. Identify the entity.
2. Identify the relationship among entities.
3. Draw ER diagram.
4. Write relational schema.

Entity Sets: An entity is a "thing" or "object" in the real world that is distinguishable from all other objects.
An entity set is a set of entities of the same type that share the same properties, or attributes.
Attributes: An entity is represented by a set of attributes. Attributes are descriptive properties possessed by each member of an entity set.

1. Simple attributes
2. Composite attributes: can be divided into subparts
3. Single – valued and multivalued attributes
4. Derived attribute: the value for this type of attribute can be derived from the values of other related attributes or entities.



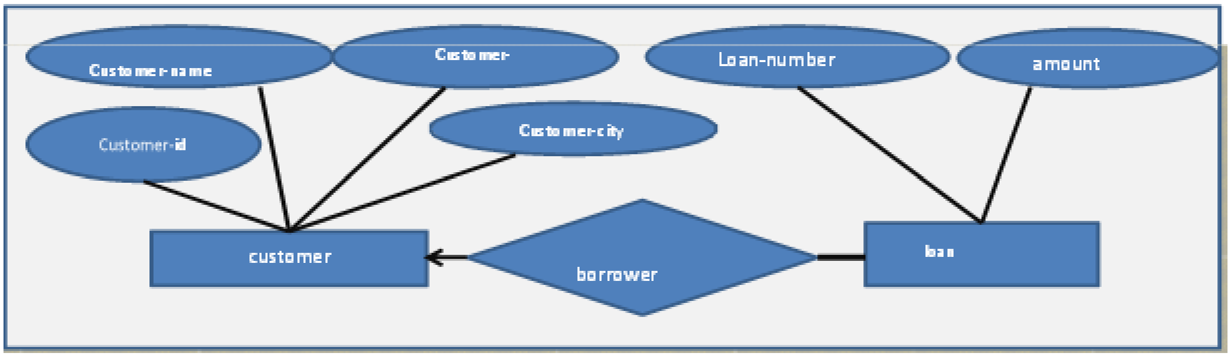
Relationship Sets: A relationship is an association among several entities.

Mapping Cardinalities:

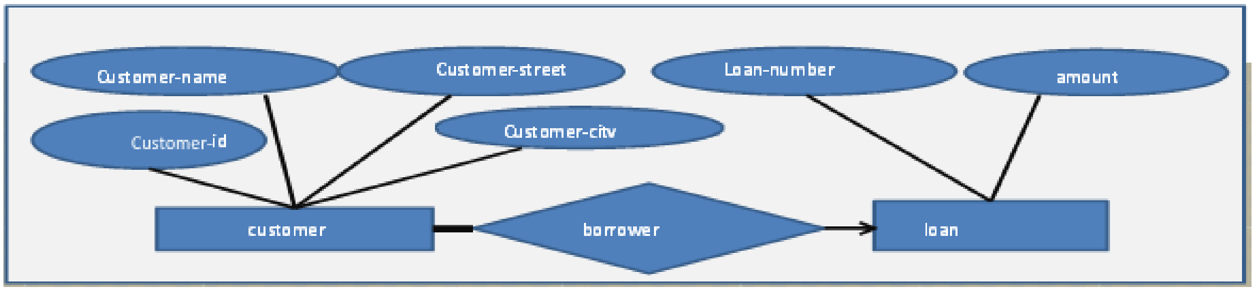
Mapping cardinalities, or cardinality ratios, express the number of entities to which another entity can be associated via a relationship set. Mapping cardinalities are most useful in describing binary relationship sets, although they can contribute to the description of relationship sets that involve more than two entity sets.

Mapping Type:

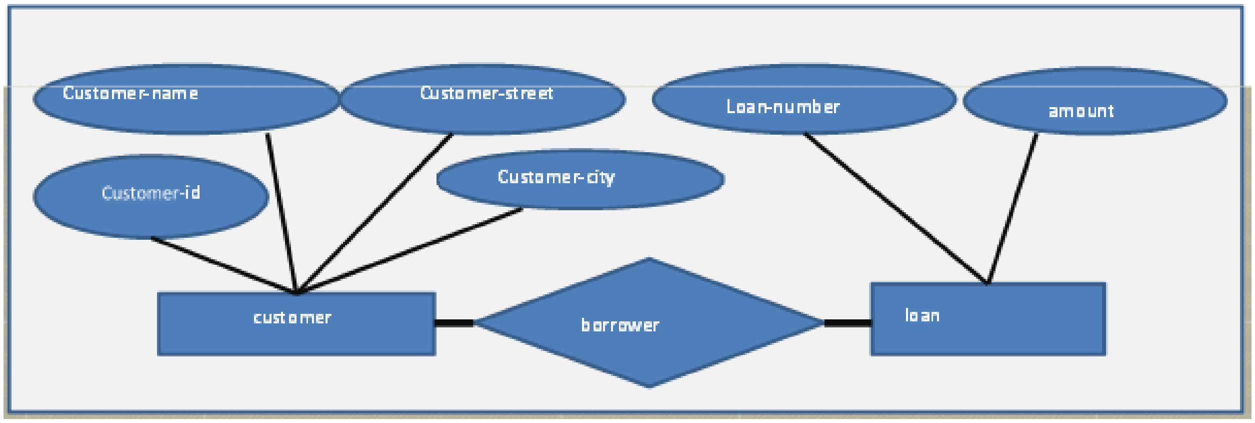
1. **One to one:** An entity in A is associated with at most one entity in B, and an entity in B is associated with at most one entity in A.
2. **One to many:** An entity in A is associated with any number of entities in B. An entity in B, however, can be associated with at most one entity in A.



3. **Many to one:** An entity in A is associated with at most one entity in B. An entity in B, however, can be associated with any number of entities in A.



4. **Many to many:** An entity in A is associated with any number of entities in b, and an entity in B is associated with any number of entities in A.



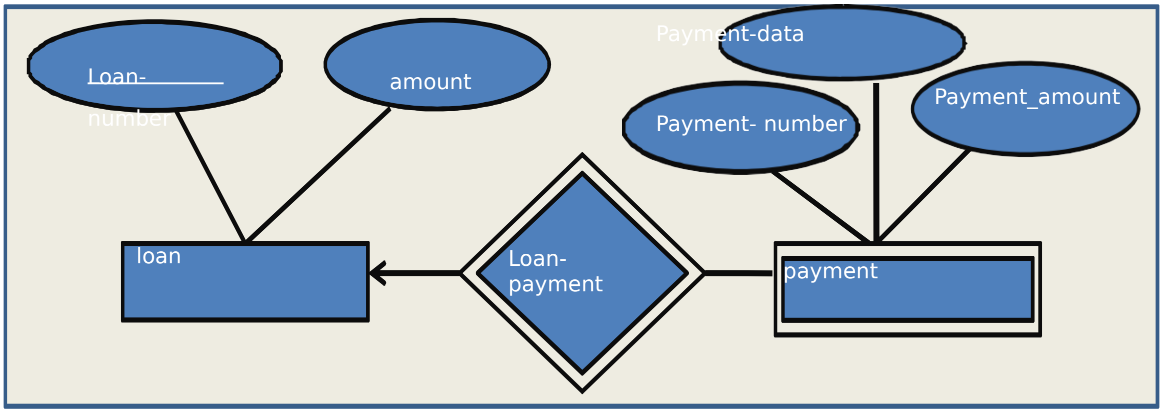
Constraints:

- **Key constraints** (Relational/referential Integrity)
  - Primary key *(unique & not null)*
  - Foreign key *(referential integrity b/w two table)*
  - Composite key *(more than one primary key)*
  - Candidate key *(multiple attributes are capable of identify the records uniquely)*
  - Alternate key *(remaining candidate key, after choosing primary key)*
- **Integrity constraints** (Domain constraints)
  - Unique constraints *(similar to primary key)*
  - Check constraints *(used to match pattern or range)*
  - Default constraints *(used to specify a fixed value)*

Entity-Relationship Diagram

- Components:
  - **Rectangles,** which represents entity sets.
  - **Ellipses,** which represent attributes.
  - **Diamonds,** which represents relationship sets.
  - **Lines,** which link attributes to entity sets and entity sets to relationship sets
  - **Double ellipses,** which represent multivalued attributes
  - **Dashed ellipses,** which denote derived attributes
  - **Double lines,** which indicate total participation of an entity in a relationship set
  - **Double rectangles,** which represent weak entity sets

Weak Entity Sets: An entity set may not have sufficient attributes to form a primary key. Such an entity is termed a weak entity set. An entity set that has a primary key is termed a strong entity set.



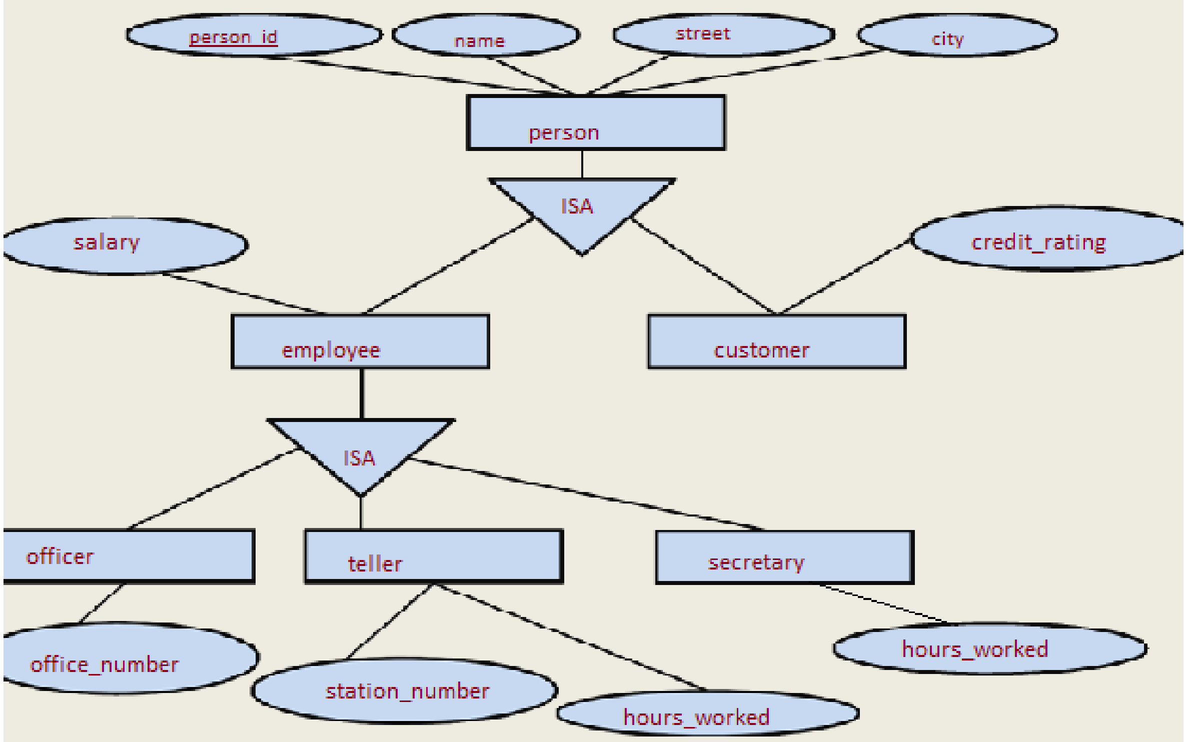
Specialization and Generalization:

The process of designing subgroupings within an entity set is called specialization. It is depicted by a triangle component labeled **ISA.** The label ISA stands for "is a".

Specialization and generalization define a containment relationship between a higher-level entity set and one or more lower-level entity sets. Specialization is the result of taking a subset of a higher-level entity to form a lower level entity set. Generalization is the result of taking the union of two or more disjoint entity set to provide a higher-level entity set. The attributes of higher-level entity sets are inherited by lower-level entity sets.

Specialization: Top-down design process

Generalization: Bottom-up design process



Aggregation: Aggregation is an abstraction in which relationship sets are treated as higher-level entity sets, and can participate in relationships.

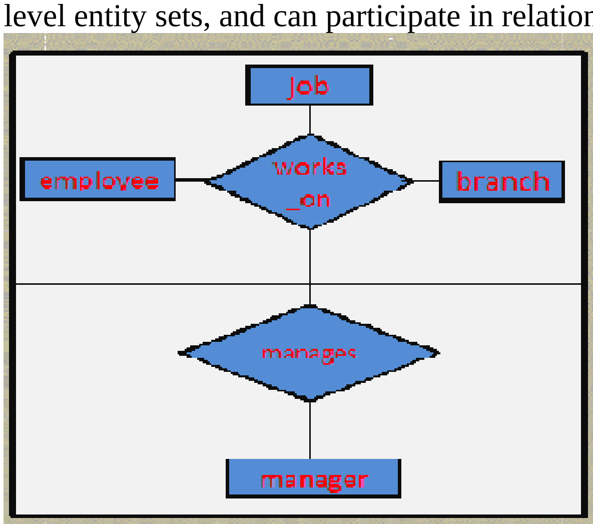


Fig : Aggregation

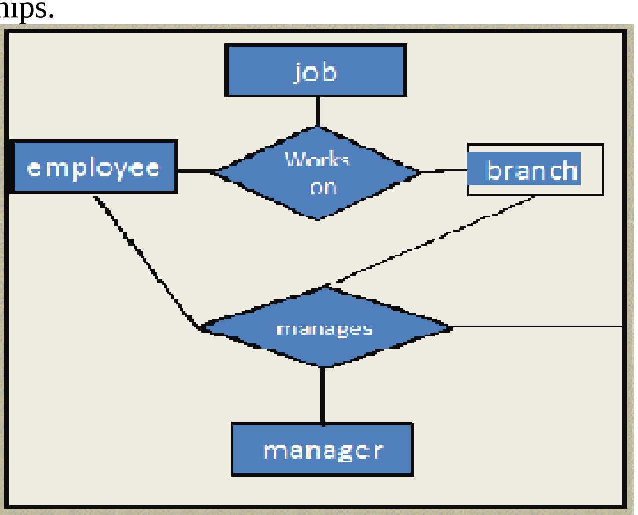


Fig : Redundant relationship