Module 2

Relational Model and E-R Modeling

Relational Model: Candidate Keys, Primary Keys, Foreign Keys - Integrity Constraints -Handling of Nulls - Entity Relationship Model: Types of Attributes, Relationships, Structural Constraints, Relational model Constraints – Mapping ER model to a relationalschema – Extended ER Model - Generalization – Specialization – Aggregations

Reference :

- ***R. Elmasri & S. B. Navathe, Fundamentals of Database Systems, Addison Wesley, 7th Edition, 2016***
- ***A. Silberschatz, H. F. Korth & S. Sudarshan, Database System Concepts, McGraw Hill,7th Edition 2019.***

Entity-Relationship (ER) model

- Entity relationship model is a popular high-level conceptual data model
- Many database design tools employ its concept
- ER model describes data as entities, relationships and attributes
- The basic object that the ER model represents is an entity
- **ER diagrams**
  - Diagrammatic notation associated with the ER model
  - Graphical representation of ER model

Database Design Process

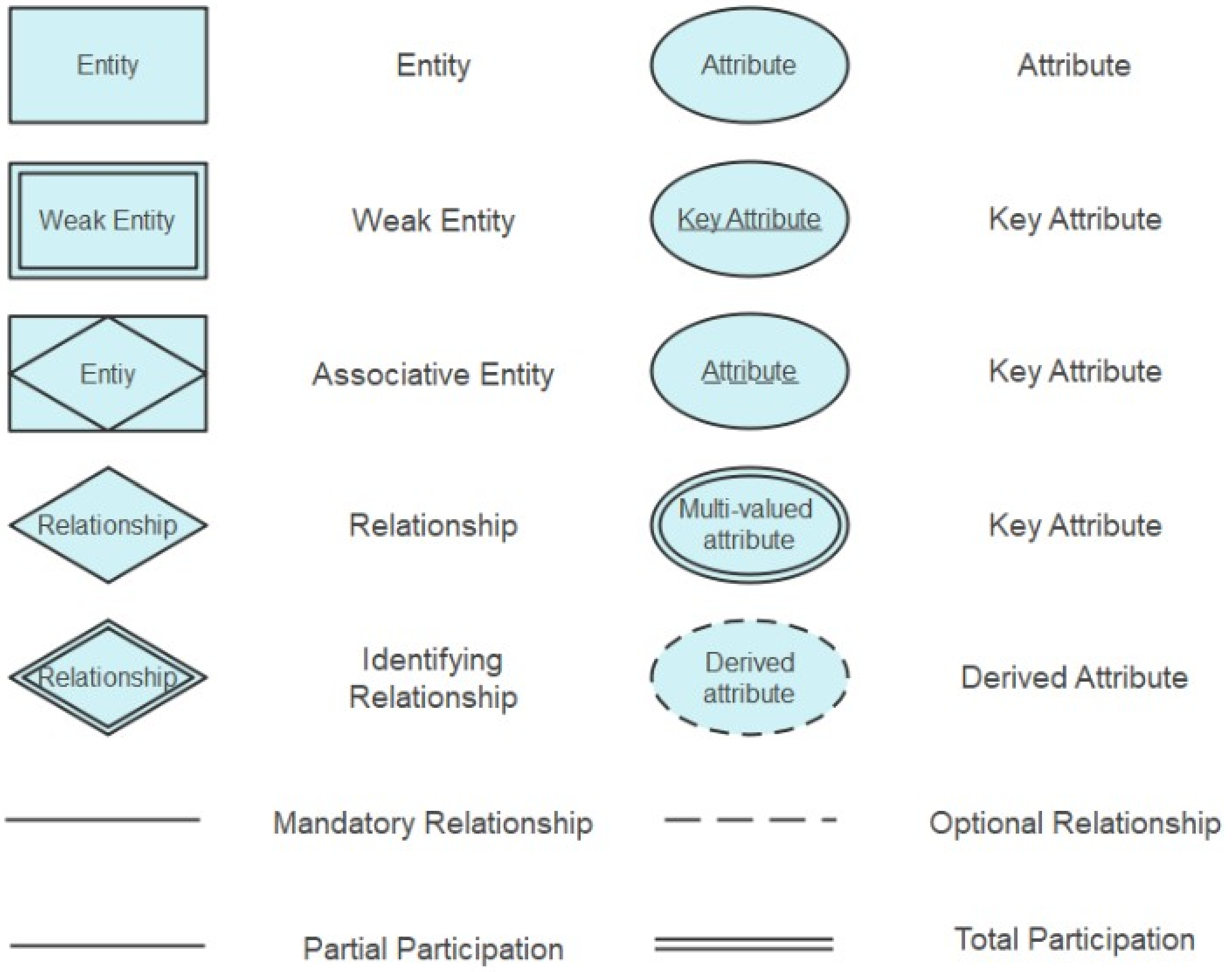
1. **Requirements Collection and Analyis**
2. **Conceptual Design → ER Model -** Captures user and system requirements clearly and abstractly

3. **Logical Design →** Translate ER to relational schema
4. **Physical Design**
5. **Implementation and Testing**

Benefits of Using ER Model

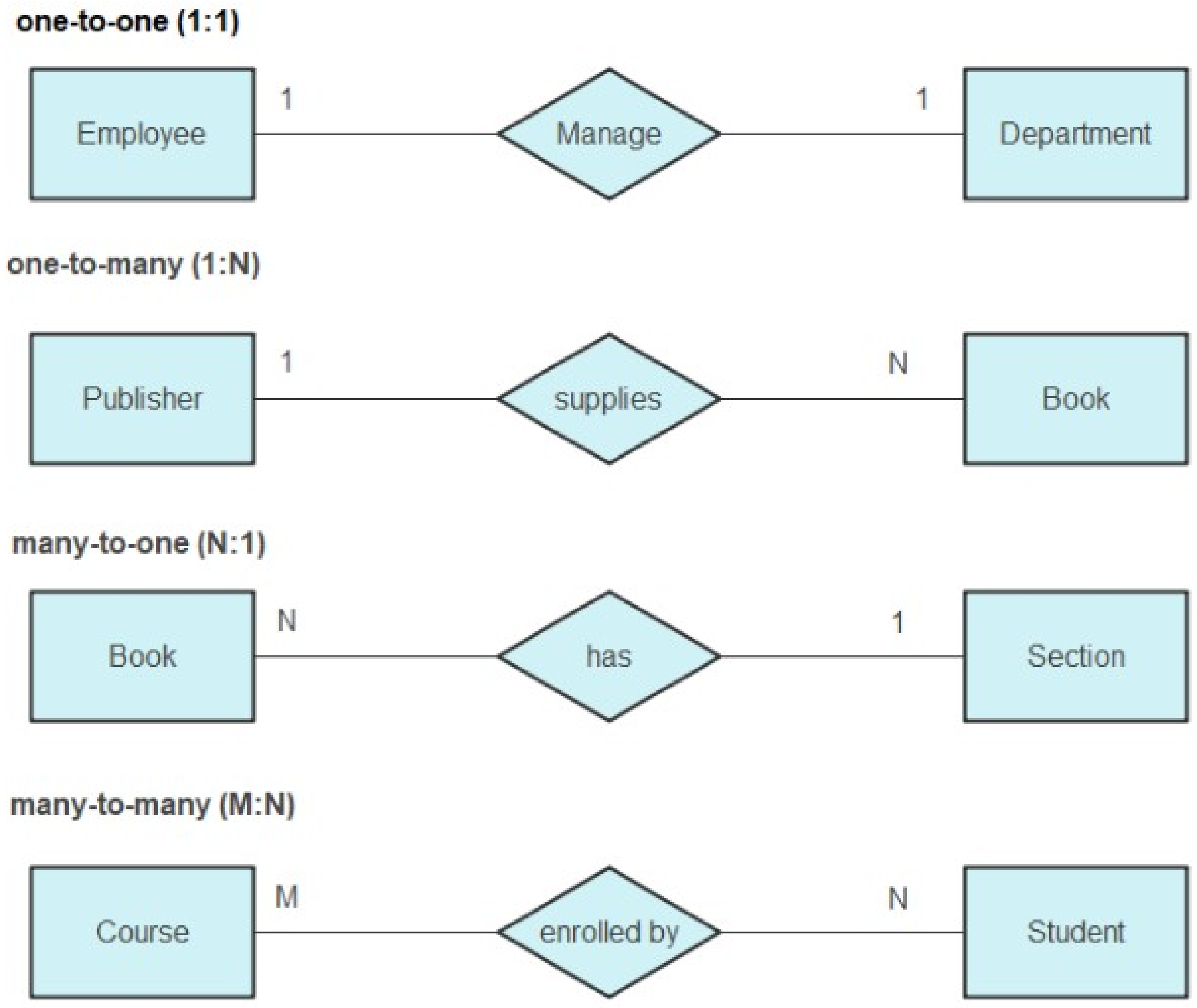
- Provides clear **visual structure** of data
- Helps with **data abstraction** before implementation
- Reduces **design errors** early in the development cycle
- Facilitates **smooth transition** to relational database systems

ERD Symbols

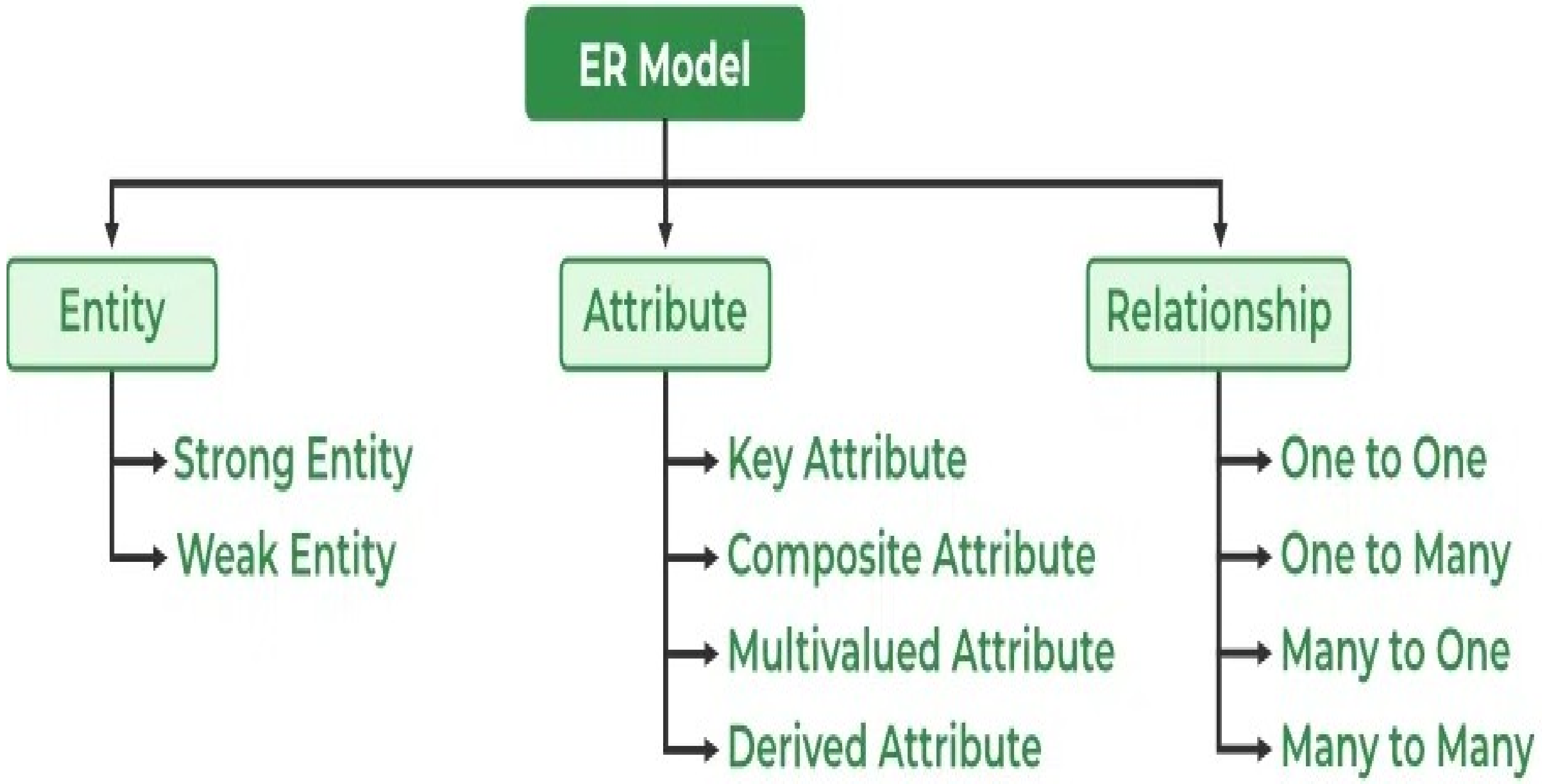


Entity
Entity
Attribute
Attribute
Weak Entity
Weak Entity
Key Attribute
Key Attribute
Entiy
Associative Entity
Attribute
Key Attribute
Relationship
Relationship
Multi-valued attribute
Key Attribute
Relationship
Identifying Relationship
Derived attribute
Derived Attribute
Mandatory Relationship
Optional Relationship
Partial Participation
Total Participation

ERD Symbols


one-to-one (1:1)
Employee
1
Manage
1
Department
one-to-many (1:N)
Publisher
1
supplies
N
Book
many-to-one (N:1)
Book
N
has
1
Section
many-to-many (M:N)
Course
M
enrolled by
N
Student

Entity-Relationship (ER) model-Components



Entity-Relationship (ER) model-Components

- **Entity**:
An objects that is stored as data such
as Student, Course or Company.
- **Attribute**:
 Properties that describes an entity such
as StudentID, CourseName, or EmployeeEmail.
- **Relationship:**
A connection between entities such as
"a Student enrolls in a Course".

Entity

- It is an **object that exists** and is distinguishable from other objects

(or)

- Entity is a "**thing**" in the real world with an independent existence.

(or)

- An entity is something that has **a distinct, separate existence**, although it need **not be a material existence**.
- In particular**, abstractions and legal functions** are usually regarded as entities

- In general, there is no presumption that an entity is animate.

1. An object with a physical existence

Example: A particular person, car, house, employee

2. An object with a conceptual existence

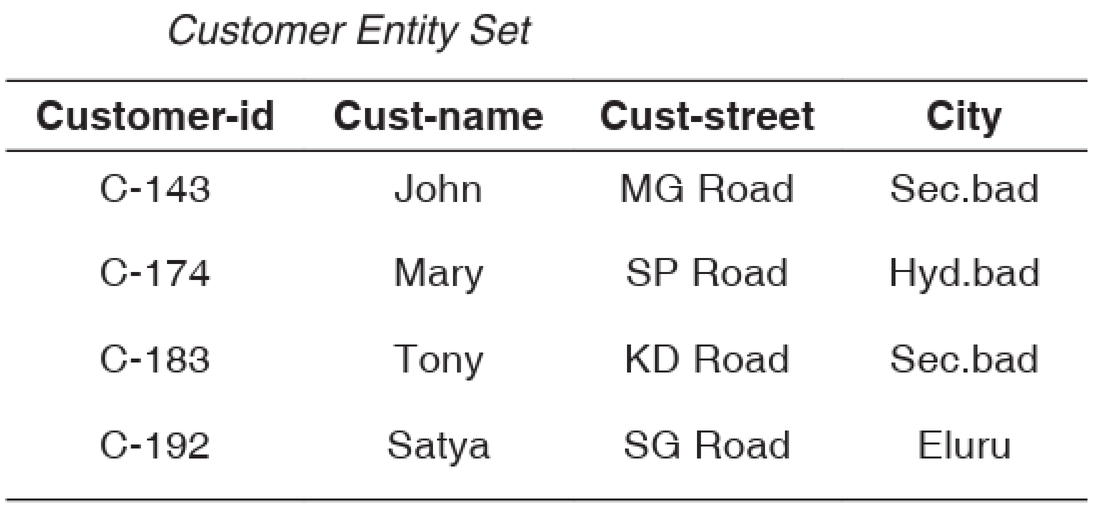
Example: A company, a job, a university course

- Each entity has attributes, the particular properties that describe it.

Example: An employee entity can be described by employee's name, age, address, salary and job

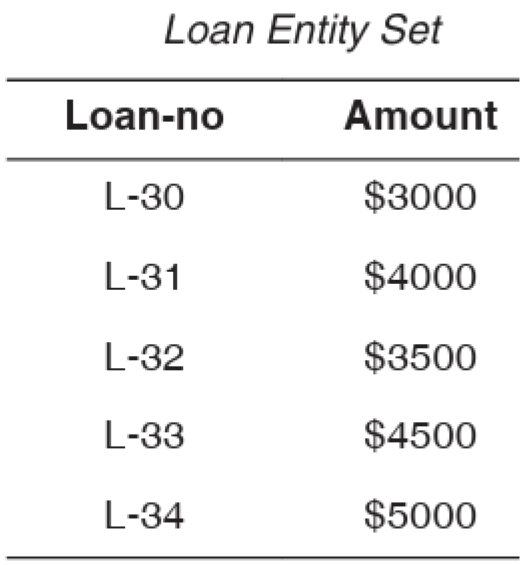
Entity Set

- Set of entities of same type that shares the same properties.
- Example: All persons, all companies etc, Entity sets of customer and loan



Customer Entity Set

Customer-id	Cust-name	Cust-street	City
C-143	John	MG Road	Sec.bad
C-174	Mary	SP Road	Hyd.bad
C-183	Tony	KD Road	Sec.bad
C-192	Satya	SG Road	Eluru



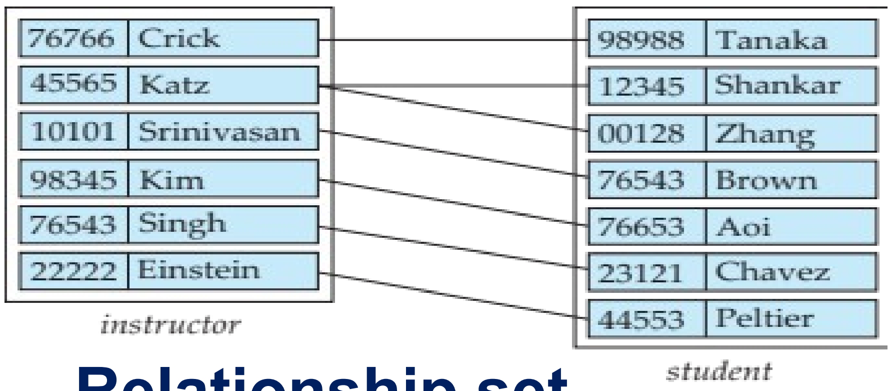
Loan Entity Set

Loan-no	Amount
L-30	$3000
L-31	$4000
L-32	$3500
L-33	$4500
L-34	$5000

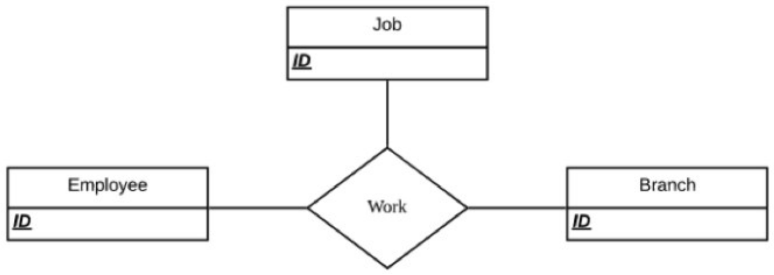
An entity is represented by a set of attributes and by a descriptive properties possessed by all members of an entity set.

Relationship Set

- A relationship is an association among several entities
- Relationship sets that involve two entity sets are binary
- Generally, most relationships in databases are binary
- Relationship sets may involve **more than two entity sets**
- Example: Employee of a bank may have responsibilities at multiple braches, with different jobs at different branches, then there is a **ternary relation between employee, job and branch.**



Relationship set



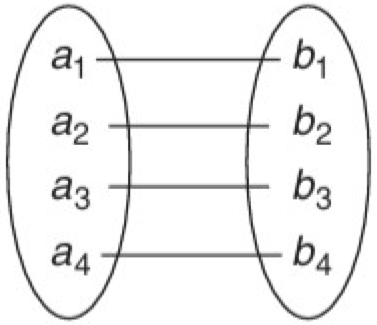
Ternary Relationship

Mapping Cardinality

- For a **binary relationship set**, mapping cardinality must be:

1. One-to-one
2. One-to-many
3. Many-to-one
4. Many-to-many

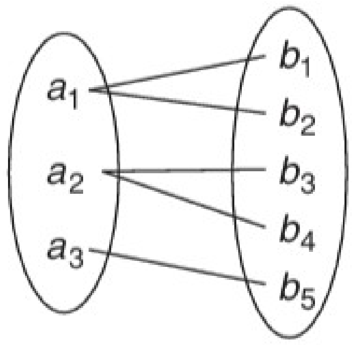
- **One-to-one:** An entity in A is associated with at most one entity in B and an entity in B is associated with at most one entity in A



One-to-one relationship set.

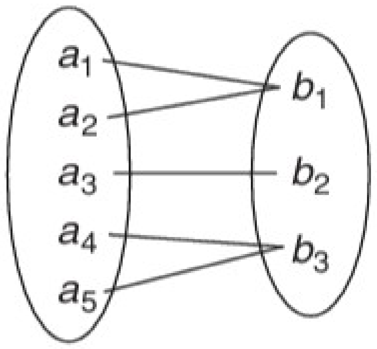
Mapping Cardinality

- **One-to-many:** An entity in A is associated with **any number** of entities in B. But an entity in B is associated with at most one entity in A



One-to-many relationship set.

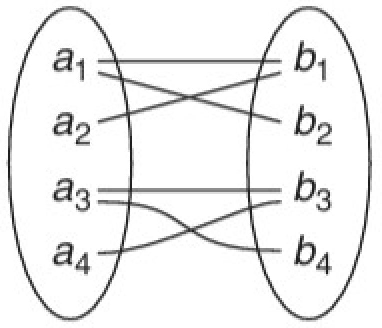
- **Many-to-one**: An entity in A is associated with at most **most one** entity in B. But an entity in B can be associated with **any number of** entities in A



Many-to-one relationship set.

Mapping Cardinality

- **Many-to-many:** An entity in A is associated with any number of entities in B. But an entity in B can be associated with any number of entities in A

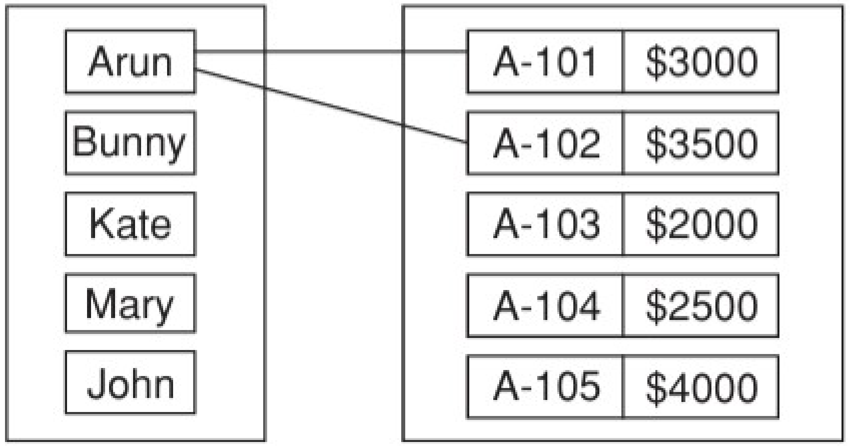


Many-to-many relationship set

Question : One customer can have multiple accounts Customer(c-Name) (Acc. no, Amount) **then what type of association exists?**

Answer?

- One-to-many Relationship Set



- many-to-one relationship is not possible.

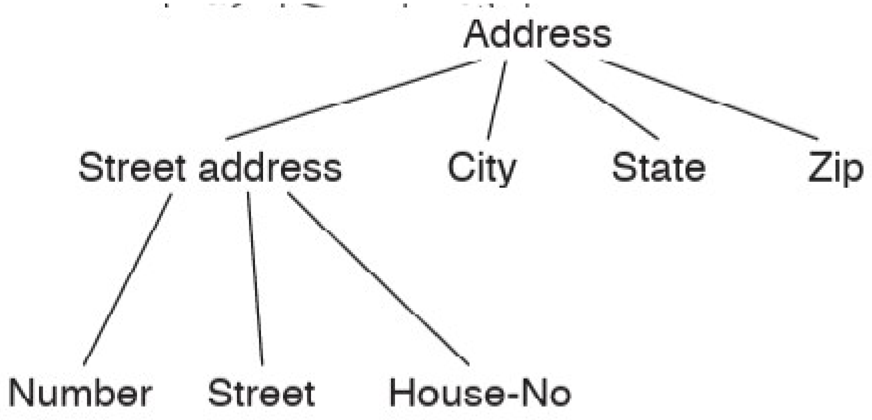
Types of Attributes

1. Simple versus composite
2. Single valued versus multivalued
3. Stored versus derived.

Composite attribute:

- Composite attributes can be divided into smaller subparts,
- Represent more basic attribute that has their own meaning
- Example: Address, Street address is a composite attribute.

Attributes that are not divisible are called **simple (or) atomic attributes.**



Types of Attributes (cont.)

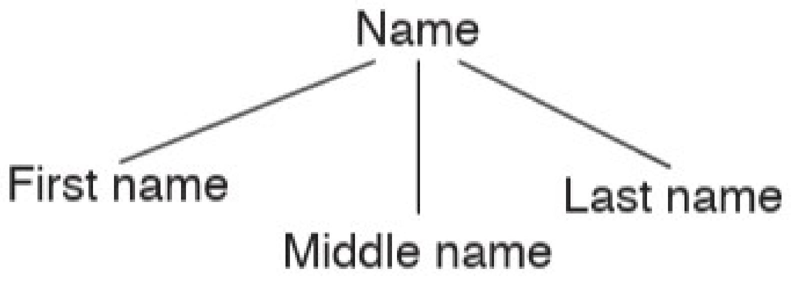
Single-valued versus multivalued attributes:

single-valued attribute

- Most attributes have a single value for a particular entity
- Example: Age is a single-valued attribute

Multivalued attributes:

- An attribute can have a set of values for the same entity.
- Name is also a multivalued attribute



Multivalued attribute.

Types of Attributes (cont.)

Stored versus derived attributes:

- Two (or) more attribute values are related.
- Example: Age can be derived from a person’s date of birth.
- The age attribute is called **derived attribute** and is said to be derivable from the DOB attribute, which is called a **stored attribute.**
- **Domain:** The set of permitted values for each attribute.
- Example: A person’s age must be in the domain {0-130}

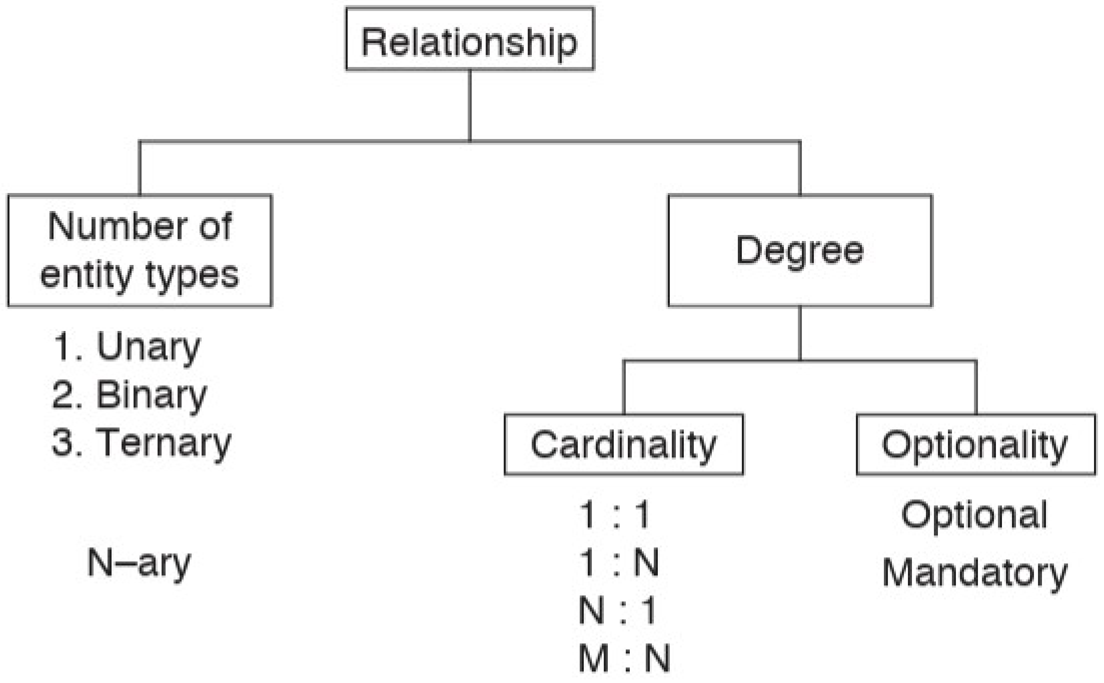
Complex and Multivalued Attributes

Example: A person can have more than one residence and each residence can have multiple phones, an attribute Address, Phone for a person can be specified as shown below.
{AddressPhone ENTITY TYPE NAME ({Phone (Areacode, phoneNumber)}, Address (StreetAddress (StreetNumber, streetName, ApartmentNumber), city, state, zip))}

- represent arbitrary nesting by grouping components of a composite attribute between parentheses () and separating the components with commas,
- displaying multivalued attributes between braces { }.

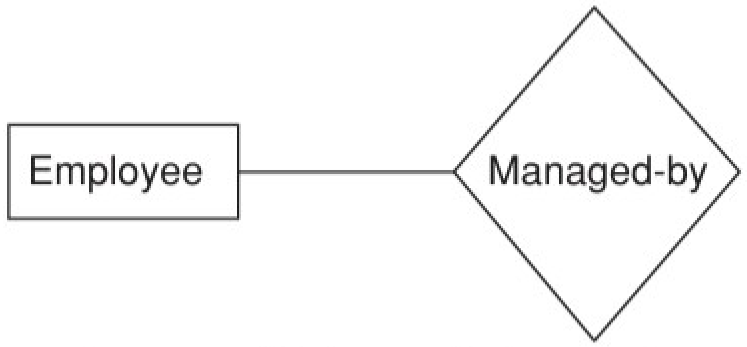
Types of relations

1. Unary relation
2. Binary relation
3. Ternary relation.
4. Quadnary relation.
5. N-ary relation



Types of relations- Unary relation

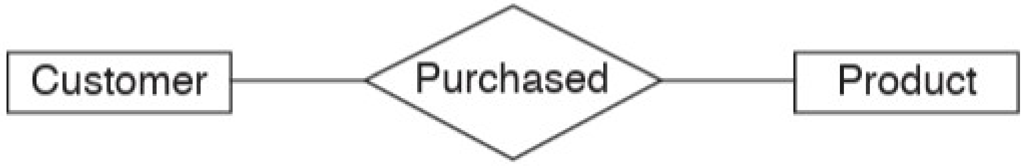
If a relationship type is between entities in a single entity type then it is called a unary relationship type.



If a relationship type is between entities in a single entity type then it is called a unary relationship type.

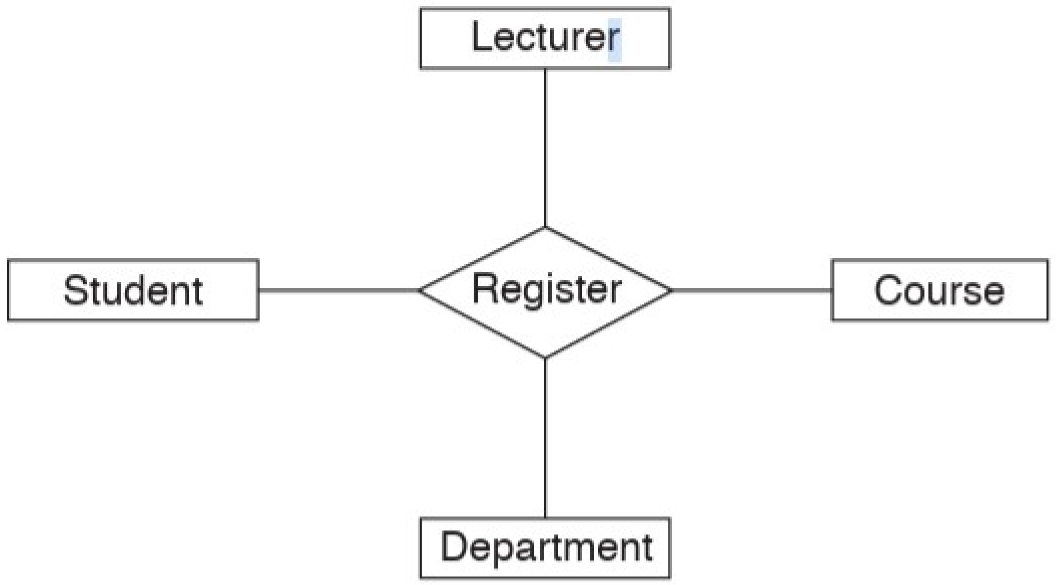
Types of relations- Binary relation

If a relationship type is between entities in one type and entities in another type then it is called a binary relation, because two entity types are involved in the relation.



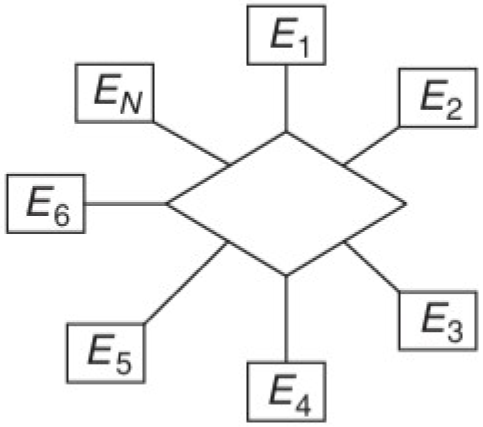
Types of relations- Quadnary relation

If a relationship type is among entities of four different types, then it is called quadnary relation.



Types of relations- N-ary relation

‘N’ number of entities will participate in a relation, and each entity can have a relation with all the other entities.



Cardinality Ratio and participation Constraints

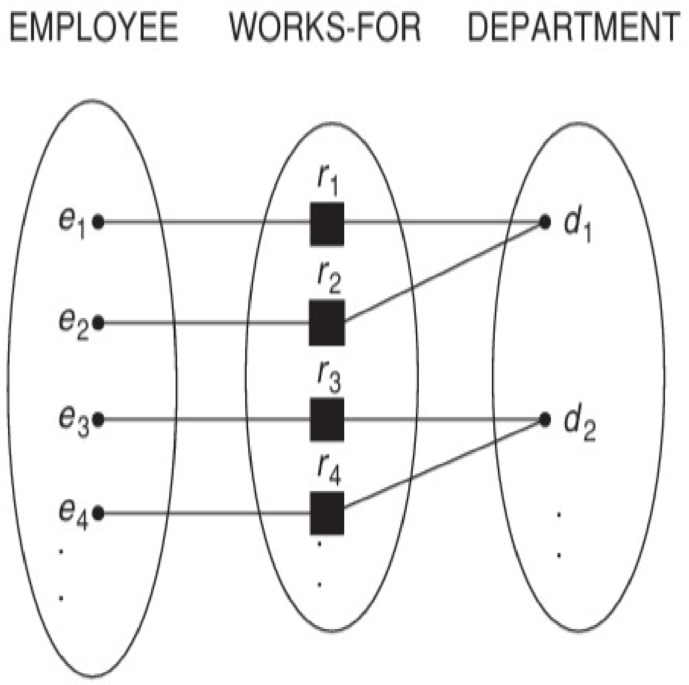
The cardinality ratio for a binary relationship specifies the maximum number of relationship instances that an entity can participate in

1. The participation constraint specifies whether the existence of an entity depends on its being related to another entity via the relationship type. This constraint specifies the minimum number of relationship instances that each entity can participate in, and is some times called the minimum cardinality constraint.
2. There are two types of participation constraints:

   a. Total participation b. Partial participation

Cardinality Ratio and participation Constraints

Example: If a company policy states that every employee must work for a department, then an employee entity can exist only if it participates (or) works for at least one department.



Total Participation (Existence Dependency):
•Every EMPLOYEE must be related to a DEPARTMENT.
•Implies no EMPLOYEE can exist without being assigned to a DEPARTMENT.
•Shown with a double line in an ER diagram.
Partial Participation:
•Not all entities of a type need to participate.
•Example: Not all EMPLOYEEs may manage a department.
•Shown with a single line in the ER diagram.