Chapter2: Database Design Using the E-R Model

-Presented by Charul Singh

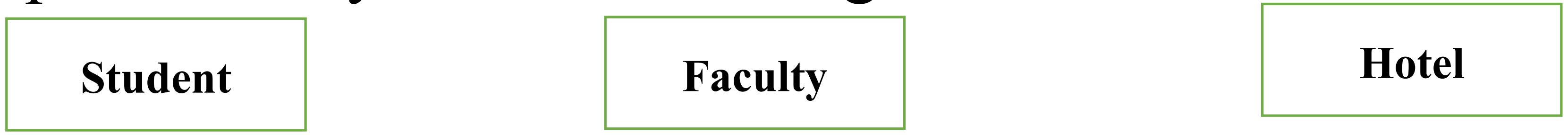
Outline

The Entity Relationship Model

- ER model was introduced by Dr.Peter Chenn in 1976
- It defines the logical view of a database
- It is used for designing database
- It works around real-world entities and the relationships among them.
- The ER data model employs three basic concepts:
    - entity sets,
    - relationship sets,
    - attributes.
- A database schema in the ER model can be represented pictorially as ER diagrams
- An ER diagram maps well into a relational schema

Entity

- An **entity** is a thing or object that exists in the real world that is distinguishable from other objects based on the values of the attributes is posses.
- Represented by means of rectangles



Types of Enities:

Tangible: Entities which physically exists in real world

Eg: Pen, Bank_locker

Intangible:Enitites which exists logically

Eg: Account

Entity Set

- An **entity set** is a set of entities of the same type that share the same properties or attributes.
  - Example: set of all persons, companies, trees, holidays

E1
E2
E3
Entity Set

- Entity sets need not to be disjoint

i.e the entity set Employee (all employees of a bank) and the entity customer (all customers of the bank) may have members in common.

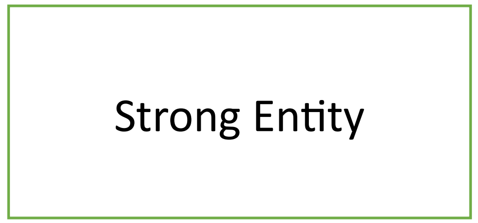
Types of Entities

- There are two types of entities in ER Diagram:

1. Strong Entity
2. Weak Entity

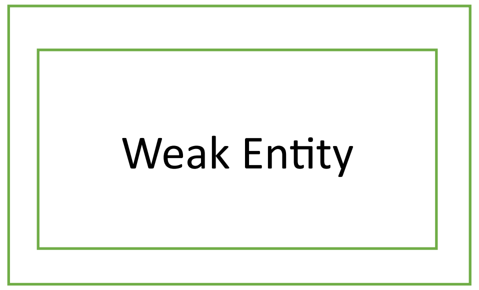
Strong Entity

- The strong Entity always have a primary key
- Its existence is not dependent on any other entity i.e it is independent of other entity
- A set of strong entities is known as strong entity set.
- Strong Entity is represented by a single rectangle



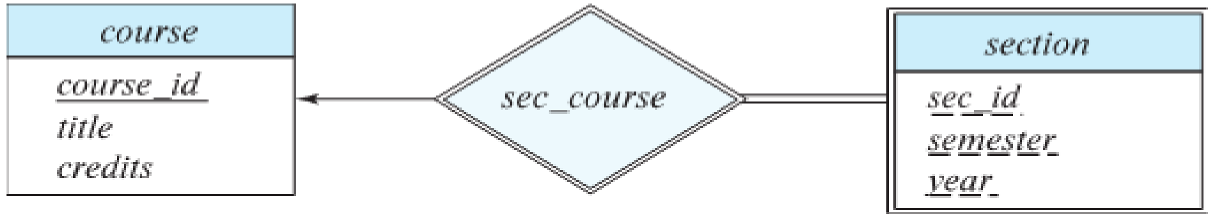
Weak Entity

- The weak entity does not have a sufficient attributes to form a primary key i.e. Weak Entity do not have a primary key
- A weak entity is dependent on a strong entity to ensure its existence
- A set of weak entities is known as weak entity set
- Weak entity is represented by double rectangle.

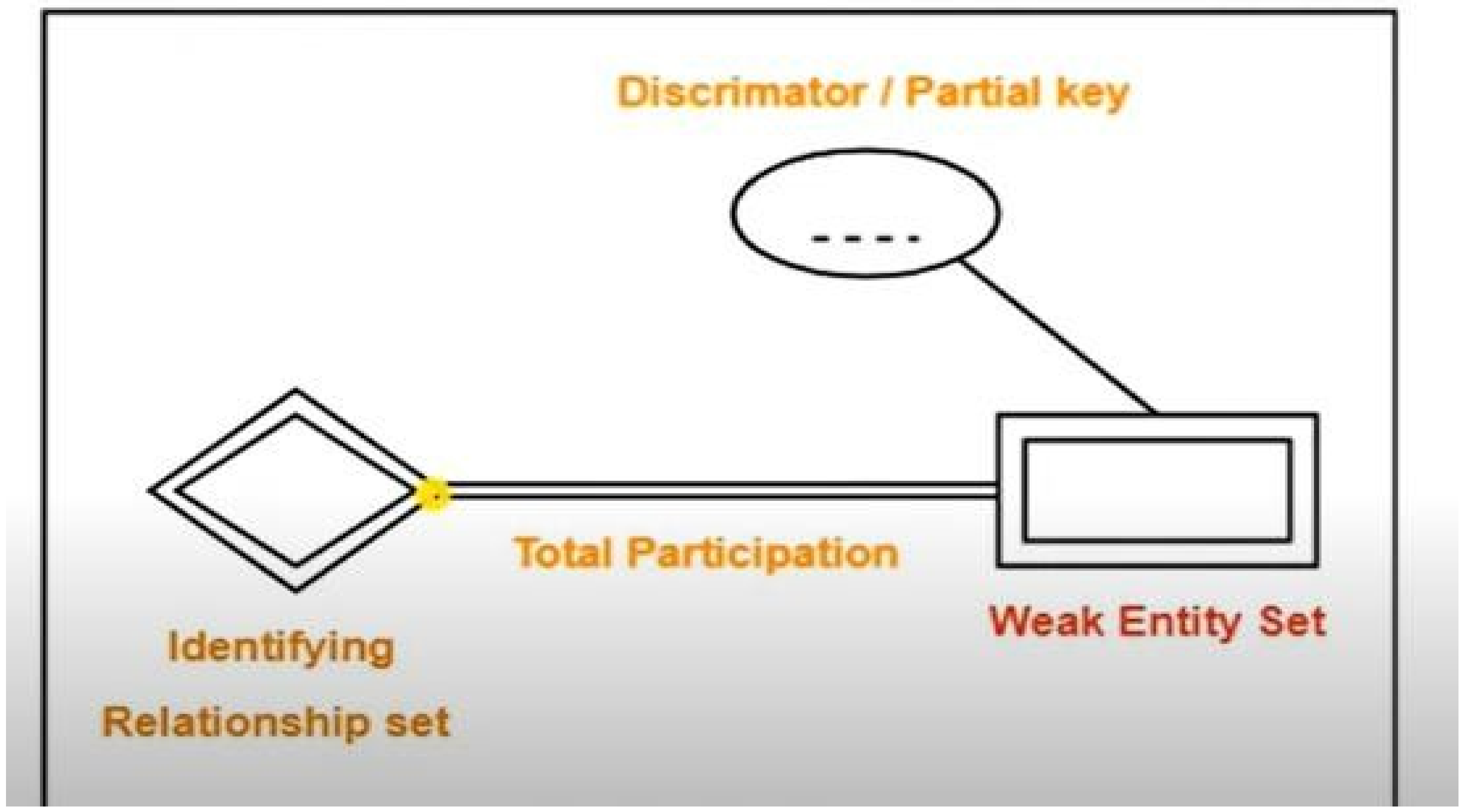


Expressing Weak Entity Sets

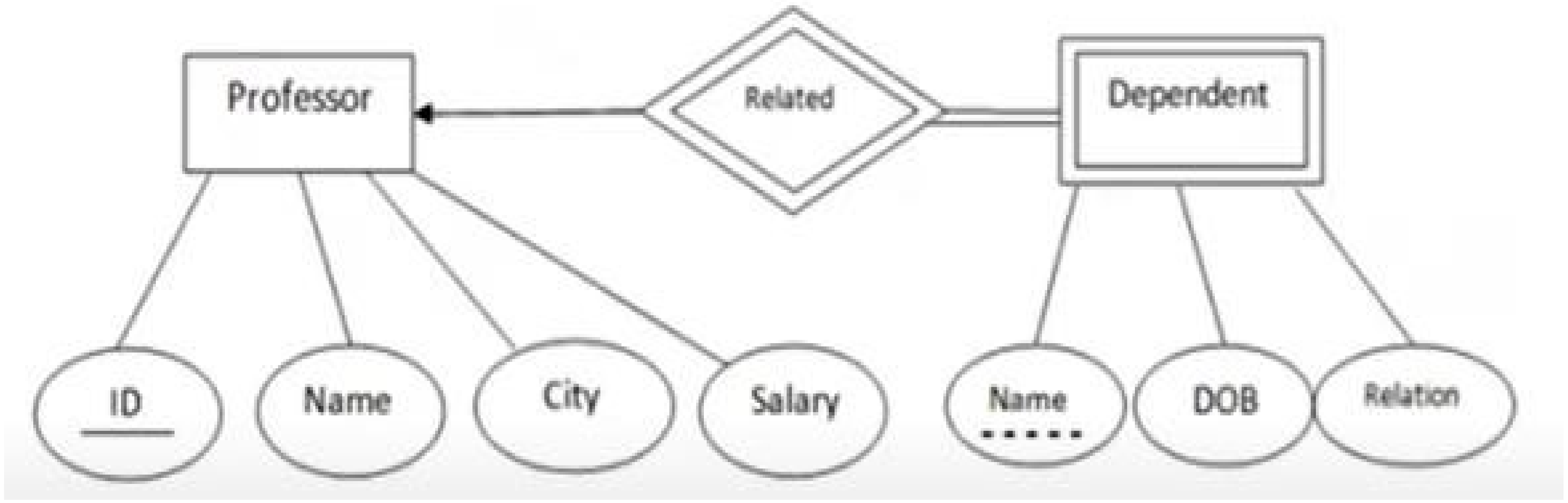
- In E-R diagrams, a weak entity set is depicted via a double rectangle.
- We underline the discriminator of a weak entity set with a dashed line.
- The relationship set connecting the weak entity set to the identifying strong entity set is depicted by a double diamond.
- Primary key for *section* – (*course_id, sec_id, semester, year*)



Representation



Example


Professor
Related
Dependent
ID
Name
City
Salary
Name
DOB
Relation

Strong Entity

- Strong Entity always has primary key
- Strong Entity is not dependent of any other entity
- Strong Entity is represented by single rectangle
- Two strong Entity Relationship is represented by single diamond.
- Strong Entity have either total participation or not.

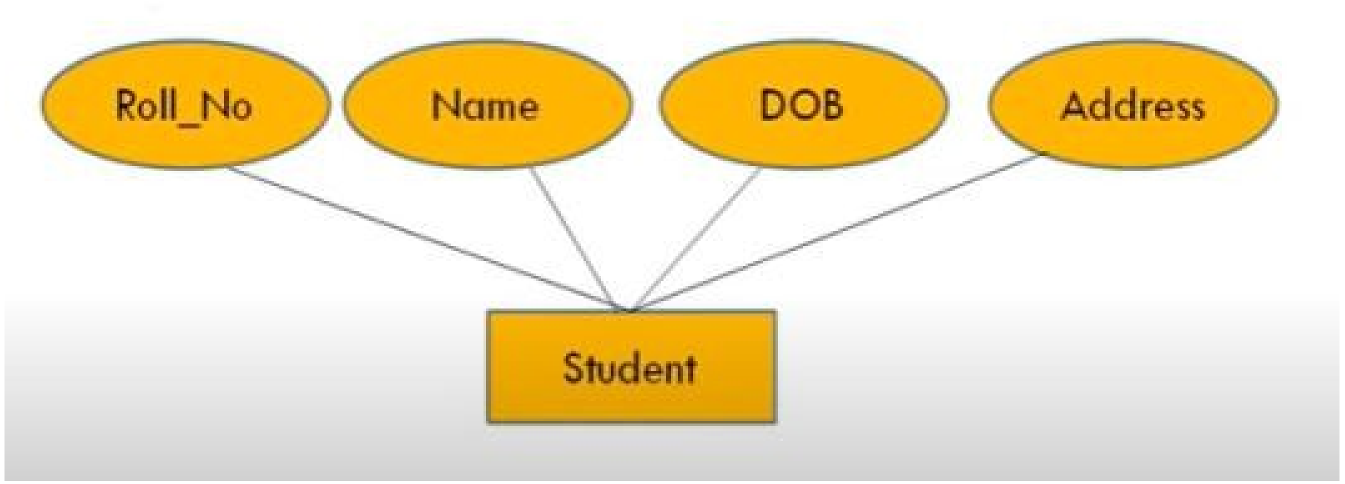
Weak Entity

- Weak Entity has partial key or discriminator key.
- Weak Entity is depends on strong entity
- Weak Entity is represented by double rectangle
- The relation between one strong & one weak entity is represented by double diamond.
- Weak Entity always has total participation.

Attributes

- It describes the property /characteristics of an entity
- Example: a Teacher entity have teacher_id,t_name,t_address etc as attribute
- For each attribute there is a set of permitted (range) values called domain
- Eg: a teachers name cannot be a numeric value. It has to be alphabetic. Also teacher's age cannot be negative..
- In ER diagram they are represented by Oval or Ellipse while in relational model by separate column.

- Schema (Entity Type)
- Student (Roll_no,Name,DOB,Address)



- Entity 1:
- 201, Rajashri chaudhari, 11-10-93,Mumbai
- Entity 2:
- 202, Jayshree jha, 14-09-92, Delhi

Attributes Types

- Attribute types:
  1. *Simple* and *composite* attributes.
  2. *Single-valued* and *multi-valued* attributes
     - E.g. multivalued attribute: *phone-numbers*
  3. *Stored & Derived* attributes
     - Derived: Can be computed from other attributes
     - E.g. *age*, given date of birth
  4. *Key attribute*

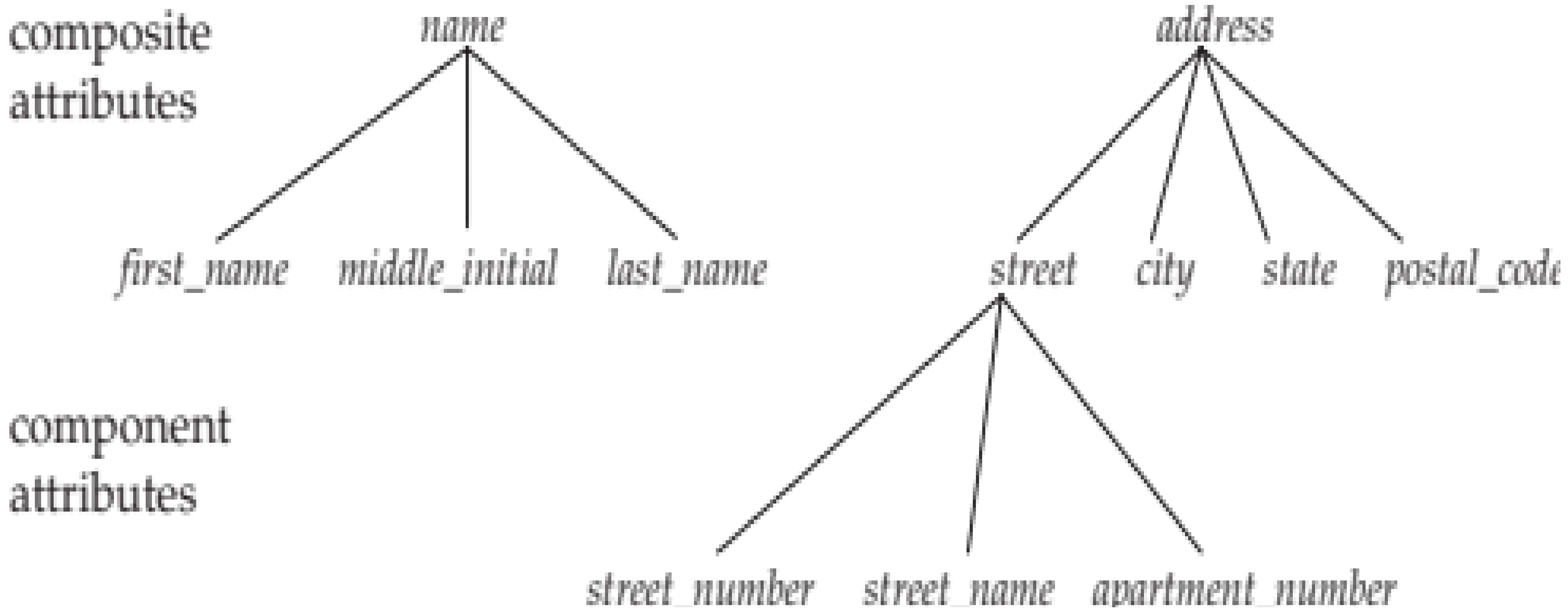
1. Simple Attributes

- They are atomic values,which cannot be divided further.
- Eg: a Birth date is an atomic value of date-month-year
- A students mobile number is an atomic value of 10 digits.
- Represented by simple oval

Composite attributes

- They are made of more than one simple attribute.
- It is divided in a tree like structure.
- Oval Coonected to oval
- Eg. A students complete name may have first_name,middle_name and last_name.

- Composite attributes allow us to divided attributes into subparts (other attributes).

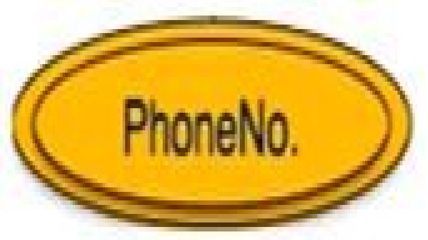


2. Single-valued attribute

- It contains only single value
- Eg: Aadhar_card_no,Moodle_id,Pan_no

Multi-valued Attribute

- It may contain more than one values.
- Represented by double-ellipse
- Eg: A person can have more than one phone number,email-address etc.

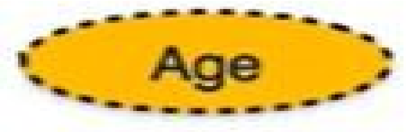


3.Stored Attribute

- Stored attributes are physically stored in the database
- Mostly all attributes are stored in database except few ones.
- Eg: DOB

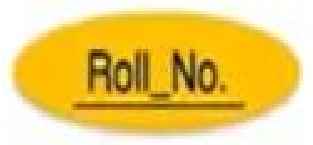
Derived Attribute

- Attributes that do not exist in the physical database, but their values are derived from other attributes present in the database.
- How value can be computed in ru time way stored database.
- They are depicted by dashed ellipse.
- E: age can be derived from date_of_birth

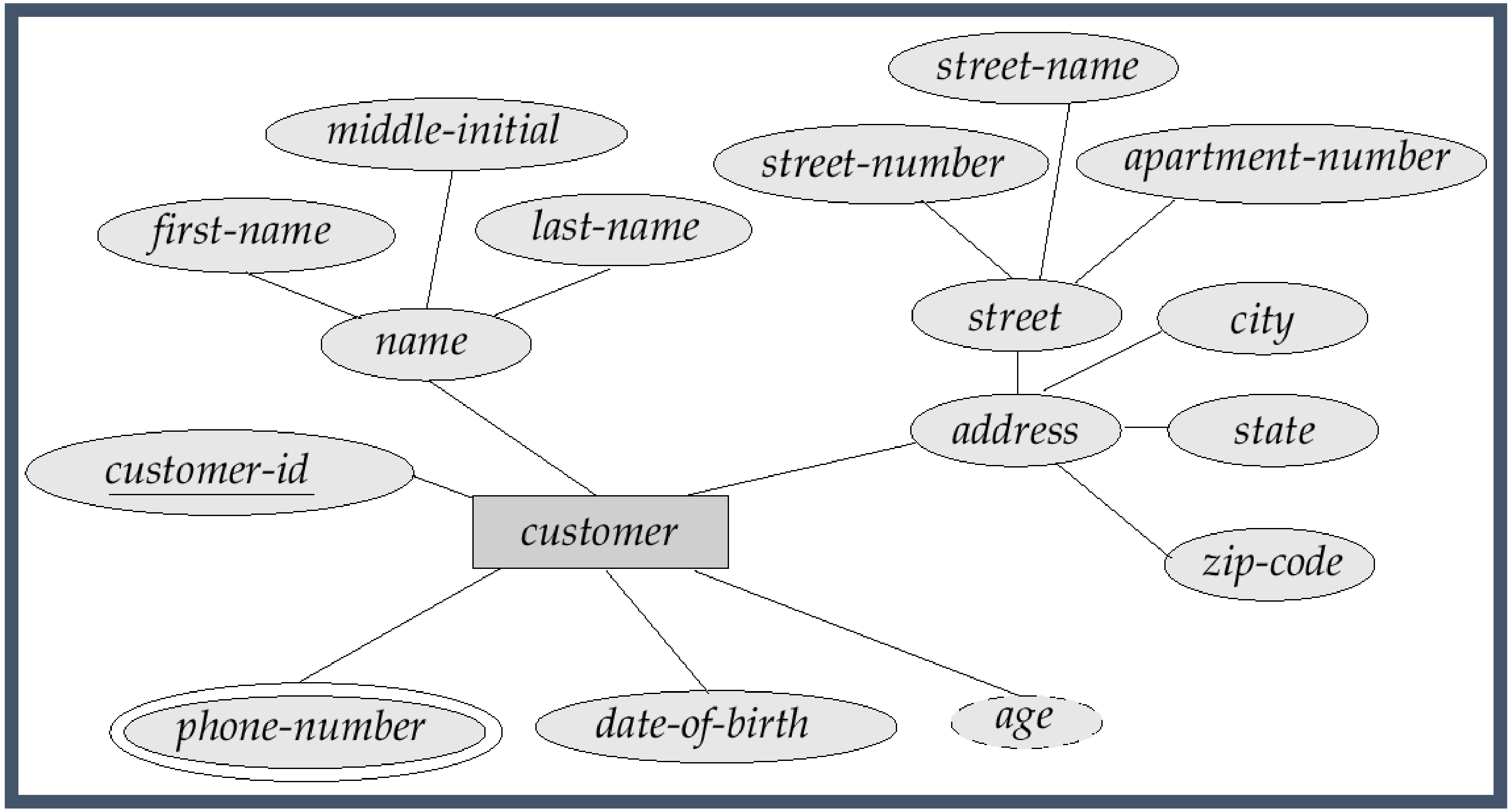


4.Key Attribute

- The attribute which uniquely identifies each entity in the entity set is called key attribute
- It represents a primary key
- It is represented by an ellipse with underlying lines.
- Eg: Roll_no will be unique for each student.



E-R Diagram With Composite, Multivalued, and Derived Attributes

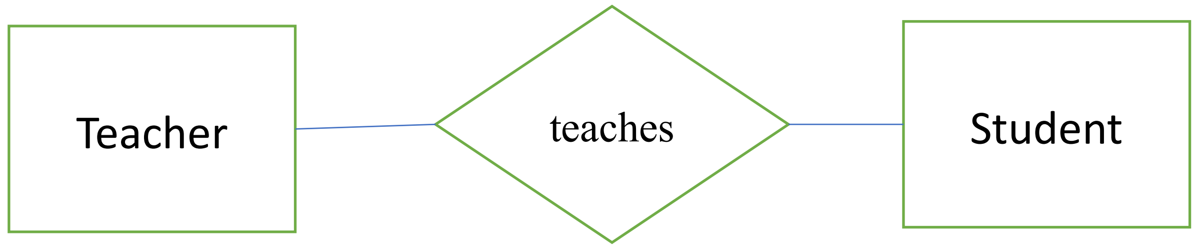


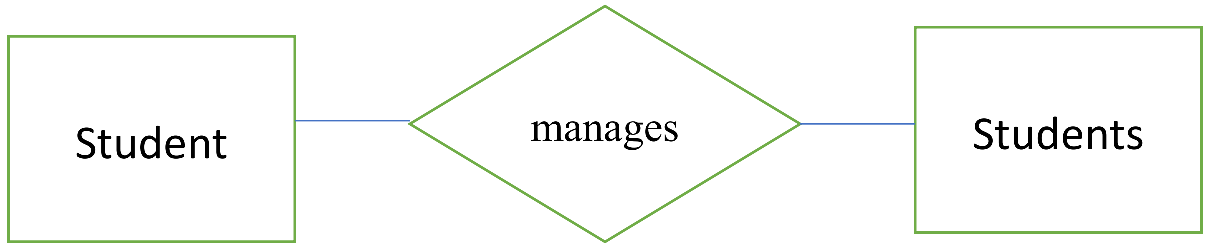
Relationship

- It is an association between two or more entities of same or different entity set.
- No Representation in ER diagram as it is an instance or data(no representation for individual relationship)

Eg:

- A teacher teaches student





Relationship Set / type

- ➢It is a set of similar type of relationship
- ➢In ER diagram represented using a diamond.
- ➢Every relationship type has 3 components

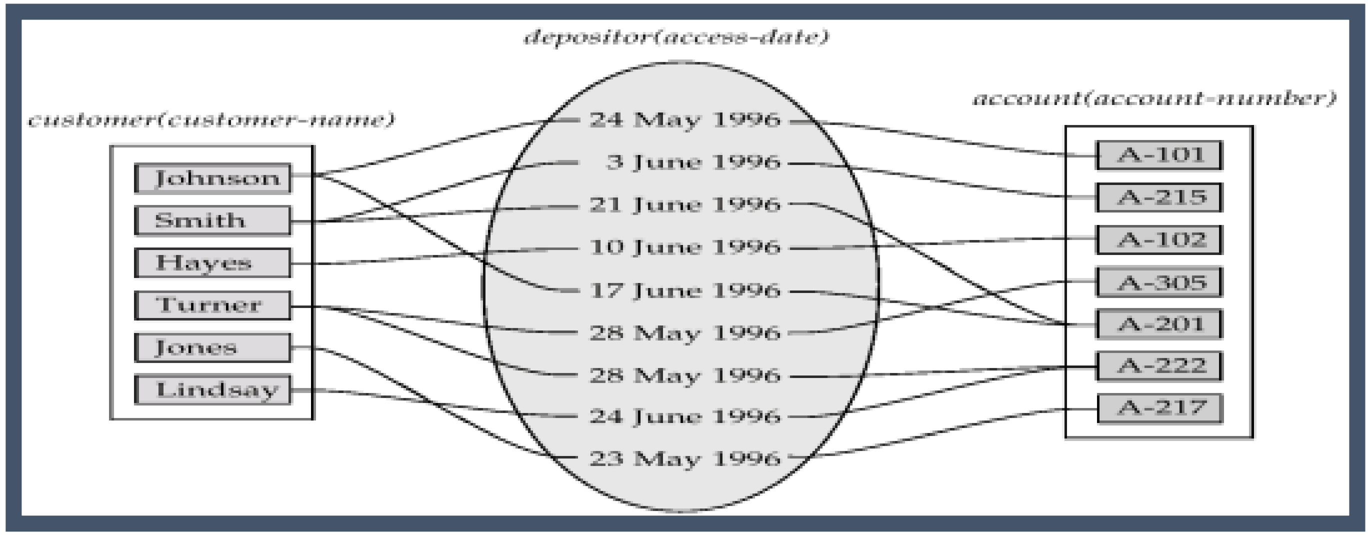
i) Name

ii) Degree

iii) Cardinality Ratio/ Participation constraints

Relationship Set (Example)

- Like entities, a relationship too can have attribute. These attribute are called descriptive attribute.
- For instance, the *depositor* relationship set between entity sets *customer* and *account* may have the attribute *access-date*



Degree of a Relationship Set

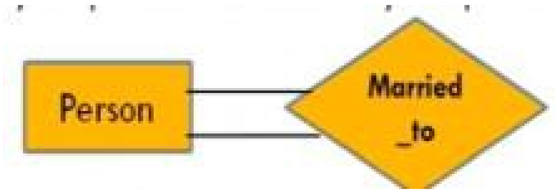
- Refers to number of entity sets that participate in a relationship set.
- ➢Unary
- ➢Binary
- ➢Ternary
- ➢N-ary
- Relationship sets that involve two entity sets are *binary* (or degree two). Generally, most relationship sets in an E-R schema are binary.
- Relationship sets may involve more than two entity sets.
  - E.g. Suppose employees of a bank may have jobs (responsibilities) at multiple branches, with different jobs at different branches. Then there is a ternary relationship set between entity sets *employee, job and branch*

Degree of a Relationship

1. Unary Relationship (degree =1)

A unary relationship is only one entity participate in a relationship., the relationship is called as unary relationship.

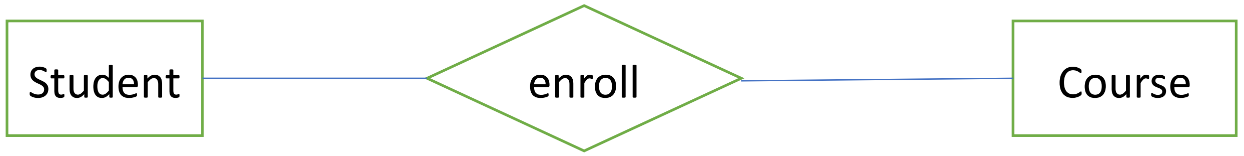
Eg: one person is married to only one person



2. Binary Relationship: (degree =2)

A binary relationship is when two entities participate in a relationship, and is the most common relationship degree

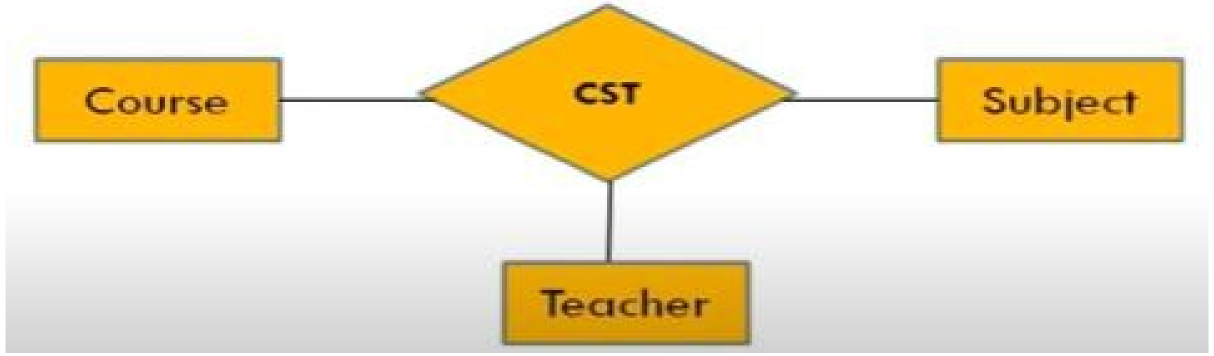
Eg: Student is enrolled in Course.



3.Ternary Relationship (degree=3)

A ternary relationships is when three entities participate in the relationship.

Eg: The University might need to record which teachers taught whuch subjects in which courses.



4. N-ary reltionships : (degree=n)

When there are n entities set participating in a relation, the reltionship is called as n-ary relationships.

Mapping Cardinalities

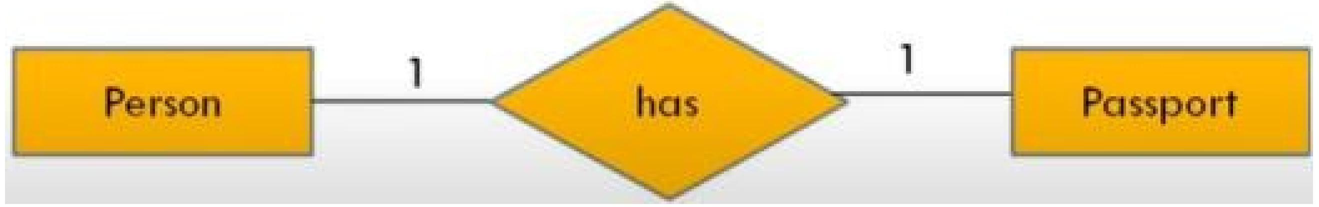
- ❑Express the number of entities to which another entity can be associated via a relationship set.
- ❑Most useful in describing binary relationship sets.
- ❑For a binary relationship set the mapping cardinality must be one of the following types:
    - ❑**One to one (1-1)**
    - ❑**One to many (1-M)**
    - ❑**Many to one (M-1)**
    - ❑**Many to many (M-N)**

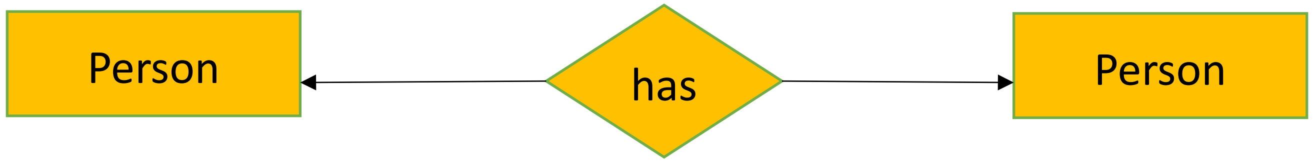
One to-One (1-1) Relationship

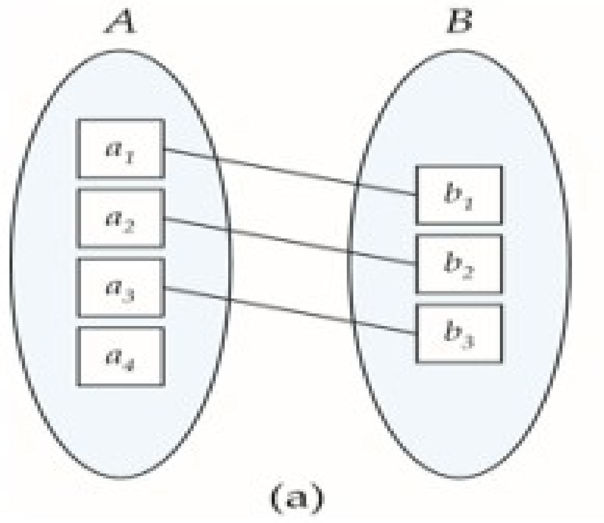
- ➢One Entity from Entity set A cab be associated with atmost one entity of entity set B and vice versa

i. Eg: A person has only one passport and a passport is given to one person

ii. Indian Citizen has Aadhar number.







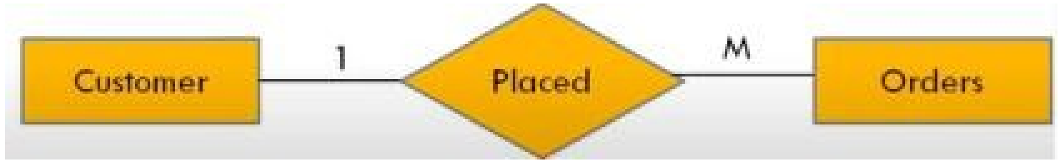
One to one

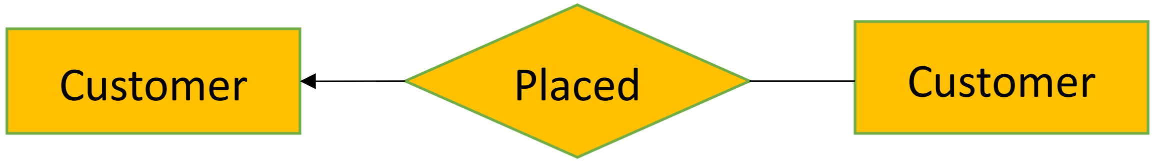
One-to-Many (1-M) Relationship

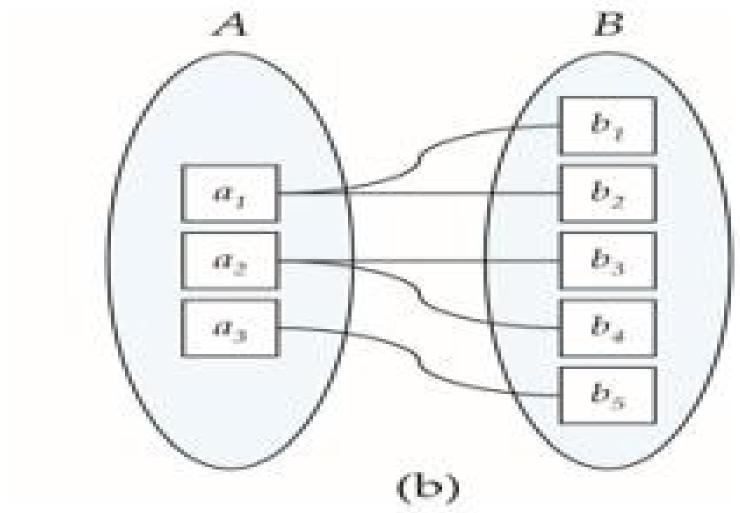
- One entity from entity set A can be associated with more than one entities of entity set B however an entity from entity B ,can be associated with at most one entity

i. Eg: A customer can place many orders but a order cannot be placed by many customers

ii. Indain citizen had mobile number



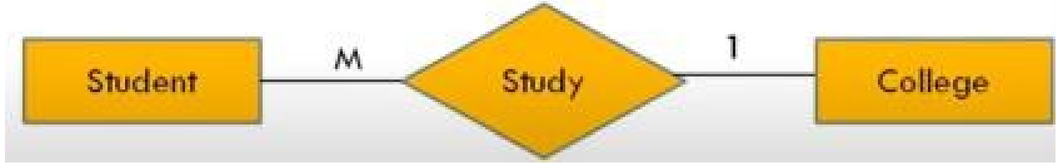


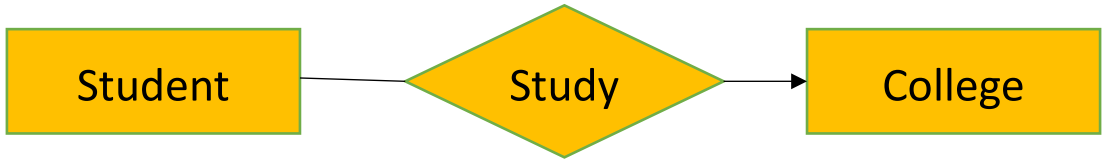


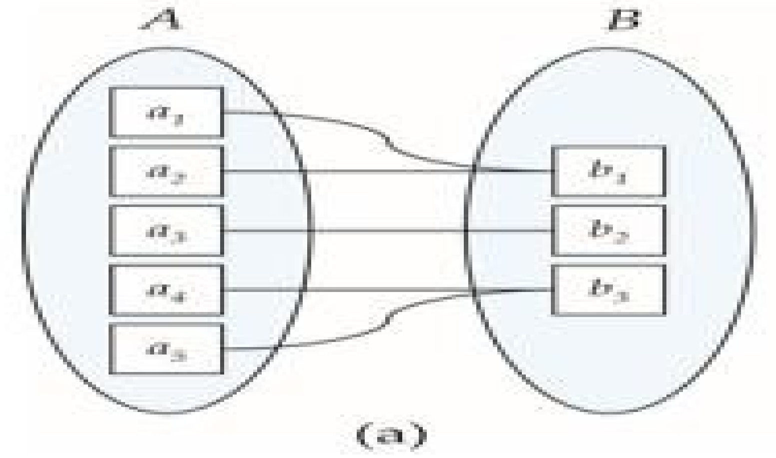
One to many

Many-to-One (M-1) Relationship

- More than one entity from entity set A can be associated with at most one entity of entity set B , however an entity from entity set B can be associated with more than one entity from entity set A.
- Eg: Many students can study in a single college but a student cannot study in many colleges at the same time.



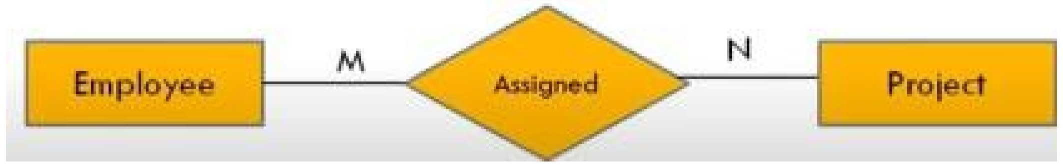


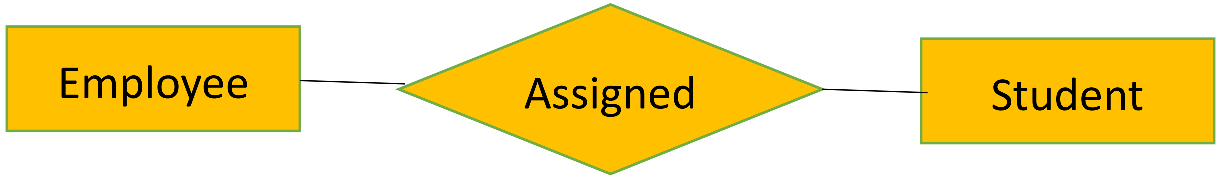


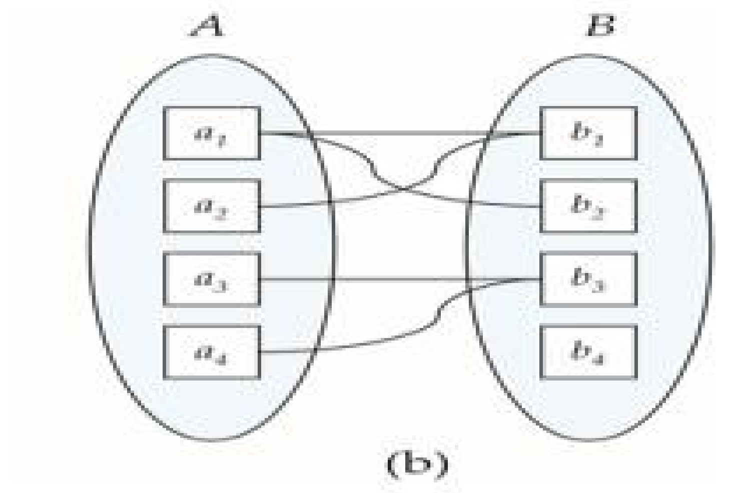
Many to one

Many-to-Many (M-N) Relationship

- One entity from A can be associated with more than one entity from B and vice versa
- Eg: An Employee can be assigned to many projects and a project can have many employees.







Many to many

Participation Constraints

- ❖It is applied on the entity participating in the relationship set
- ❑**Total Participation**
- ❑**Partial Participation**

❖Total Participation – Each entity is involved in the relationship. Total participation is represented by **double lines**

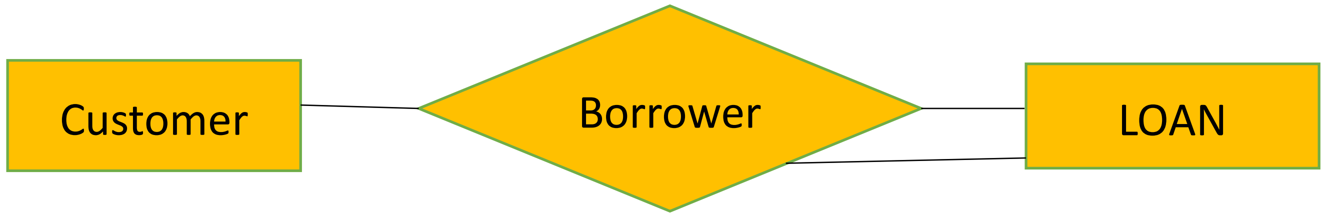
- Eg: participation of loan in borrower is total
- Every loan must have a customer associated to it via borrower

❖Partial Participation – Not all entities are involved in the relationship.

Partial Participation is represented by **single lines**

Participation of customer in borrower is partial

- A customer may have no loans.



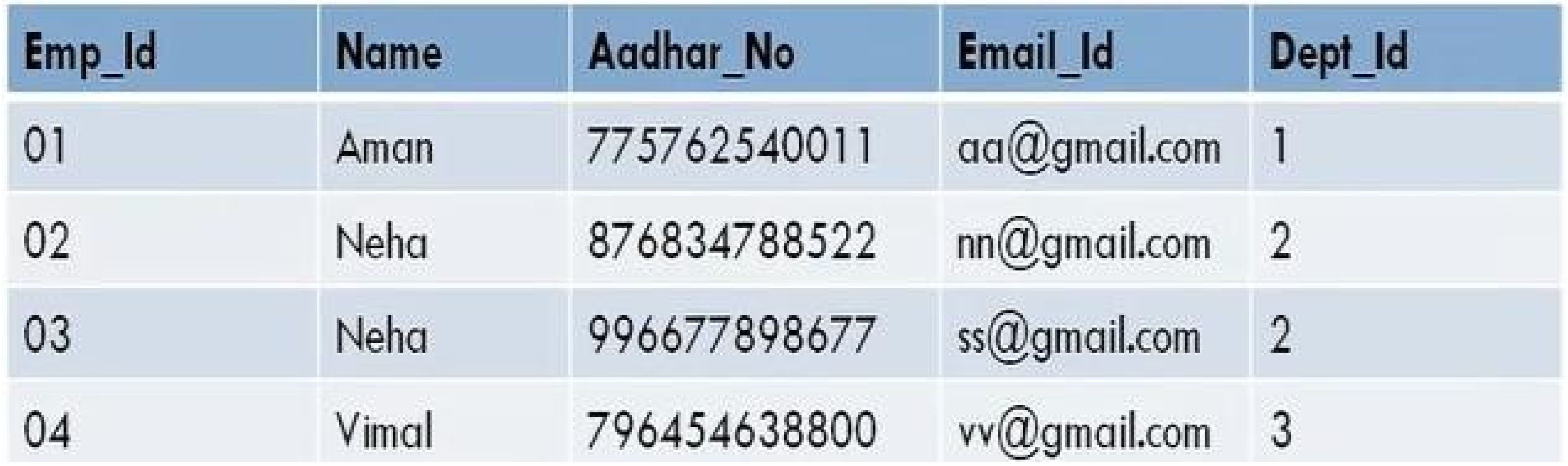
Keys in ER Diagram

➢Defn: A key is an attribute or set of attributes that uniquely identifies any record (or tuple) from the table

- Purpose:

➢Key is used to uniquely identify any record or row of data from the table

➢It is also used to establish and identify relationships between tables.



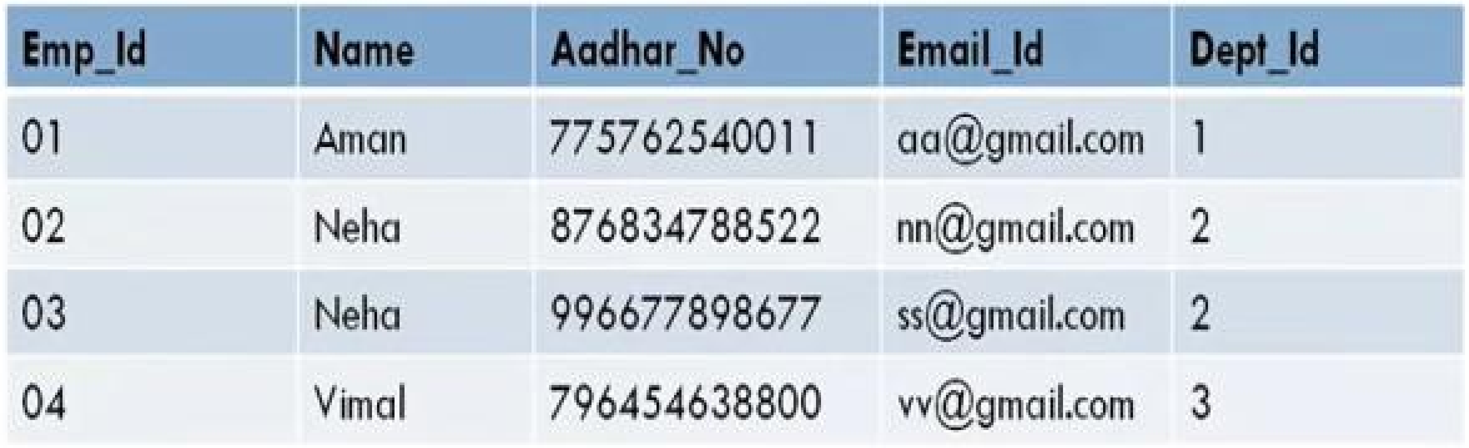
Emp_Id	Name	Aadhar_No	Email_Id	Dept_Id
01	Aman	775762540011	aa@gmail.com	1
02	Neha	876834788522	nn@gmail.com	2
03	Neha	996677898677	ss@gmail.com	2
04	Vimal	796454638800	vv@gmail.com	3

Types of keys

1. Super key
2. Candidate key
3. Primary key
4. Alternate key
5. Foreign key
6. Composite key

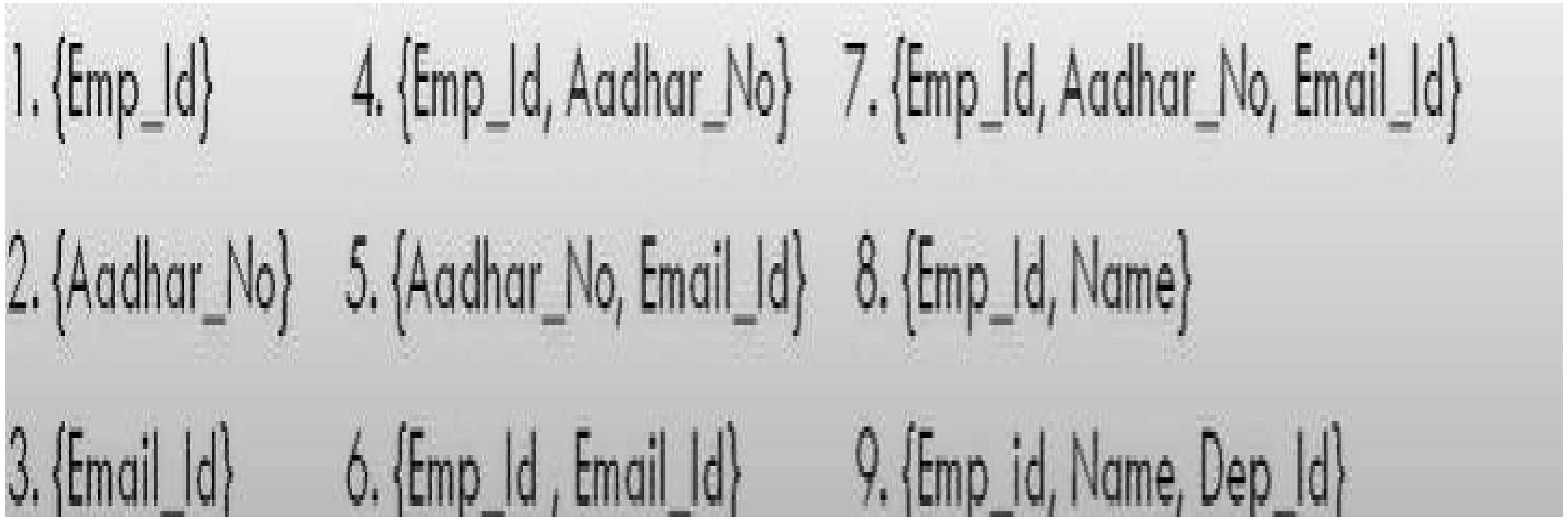
1. Super Key

- A super key is a combination of all possible attributes that can uniquely identify the rows (or tuples) in the given relation.
- Super key is a superset of a candidate key
- A table can have many super keys
- A super key may have additional attribute that are not needed for unique identity.



Emp_Id	Name	Aadhar_No	Email_Id	Dept_Id
01	Aman	775762540011	aa@gmail.com	1
02	Neha	876834788522	nn@gmail.com	2
03	Neha	996677898677	ss@gmail.com	2
04	Vimal	796454638800	vv@gmail.com	3

Super keys

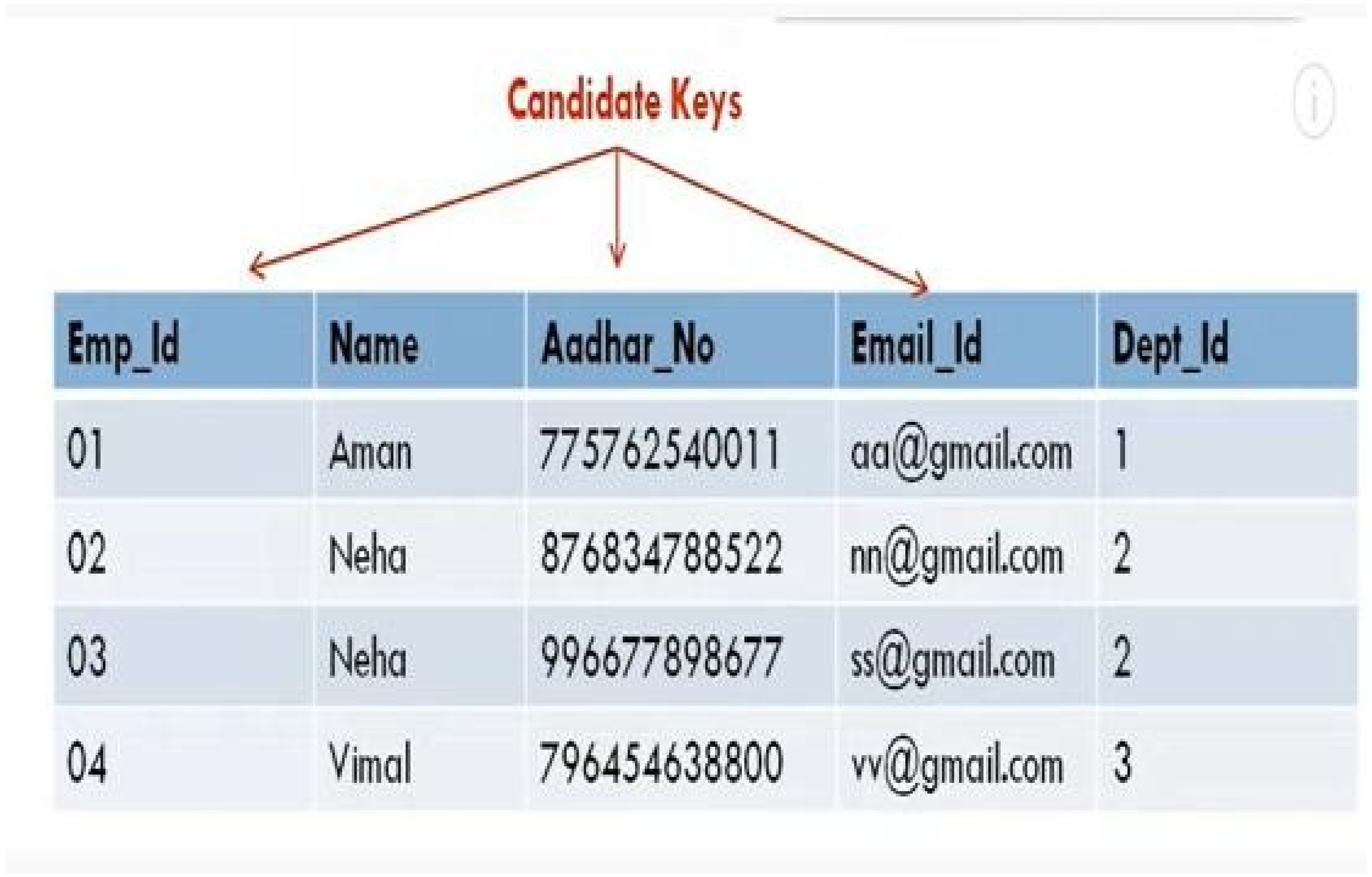


1. {Emp_Id}
2. {Aadhar_No}
3. {Email_Id}
4. {Emp_Id, Aadhar_No}
5. {Aadhar_No, Email_Id}
6. {Emp_Id , Email_Id}
7. {Emp_Id, Aadhar_No, Email_Id}
8. {Emp_Id, Name}
9. {Emp_id, Name, Dep_Id}

2. Candidate Key

- A candidate key is an attribute or set of an attribute which can uniquely identify a tuple
- A candidate key is a **minimal super key;** or a super key with no redundant attributes.
- It is called a minimal super key because we select a candidate key from a set of super key such that selected candidate key is the minimum attribute required to uniquely identify the table.
- Candidate keys are defined as distinct set of attributes from which primary key can be selected
- Candidate keys are not allowed to have NULL values

Eg:Candidate Key

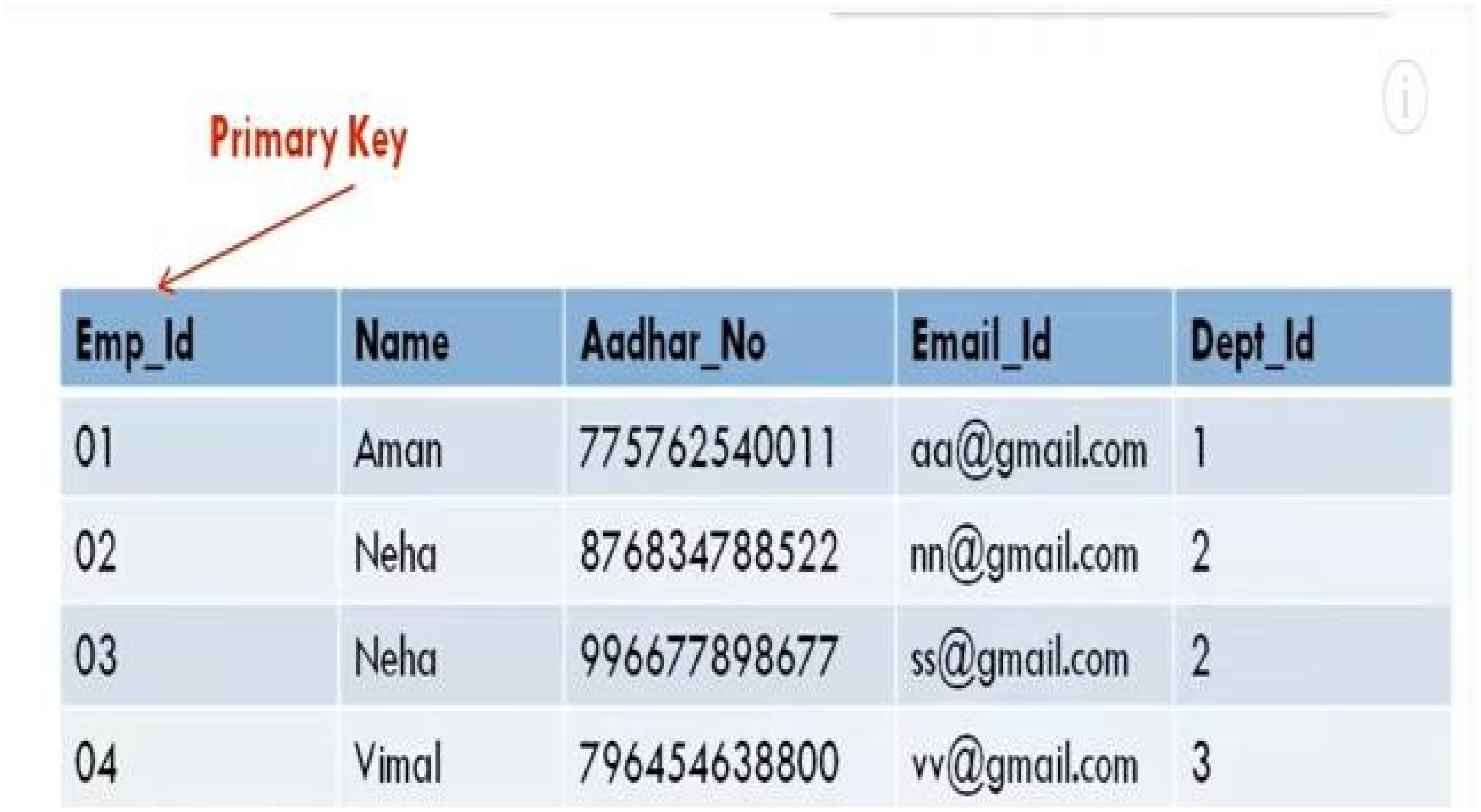

Candidate Keys

Emp_Id	Name	Aadhar_No	Email_Id	Dept_Id
01	Aman	775762540011	aa@gmail.com	1
02	Neha	876834788522	nn@gmail.com	2
03	Neha	996677898677	ss@gmail.com	2
04	Vimal	796454638800	vv@gmail.com	3

3. Primary Key

- ➢A primary key is one of the candidate key chosen by the database designer to uniquely identify the tuple in the relation
- ➢The value of primary key can never be NULL.
- ➢The value of primary key must always be unique (not duplicate).
- ➢The value of primary key can never be changed i.e no updation is possible
- ➢The value of primary key must be assigned when inserting a record.
- ➢A relation is allowed to have only one primary key.

Eg: Primary key

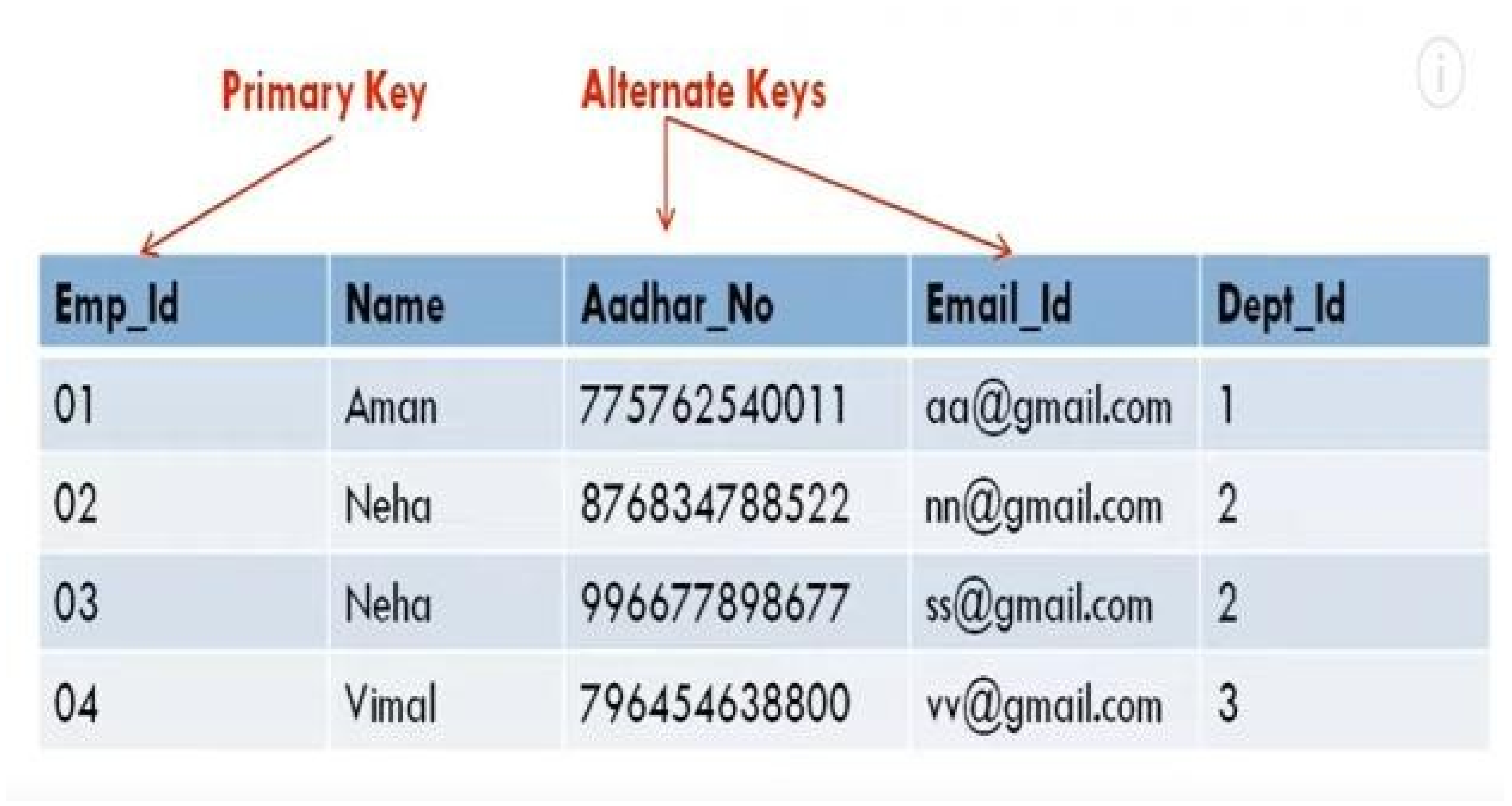

Primary Key

Emp_Id	Name	Aadhar_No	Email_Id	Dept_Id
01	Aman	775762540011	aa@gmail.com	1
02	Neha	876834788522	nn@gmail.com	2
03	Neha	996677898677	ss@gmail.com	2
04	Vimal	796454638800	vv@gmail.com	3

4. Alternate Keys

- Out of all candidate keys, only gets selected as primary key, remaining keys are also known as alternate keys.
- In the Employee table
- Emp_id is best suited for the primary key.
- Rest of the attributes like Aadhar_No, Email_id are considered as a alternate keys.

Eg: Alternate Keys


Primary Key
Alternate Keys

Emp_Id	Name	Aadhar_No	Email_Id	Dept_Id
01	Aman	775762540011	aa@gmail.com	1
02	Neha	876834788522	nn@gmail.com	2
03	Neha	996677898677	ss@gmail.com	2
04	Vimal	796454638800	vv@gmail.com	3

5. Foreign keys

- A foreign key is :

➢ A key used to link two tables together.

➢ An attribute (or set of attributes) in one table that refers to the Primary key in another table.

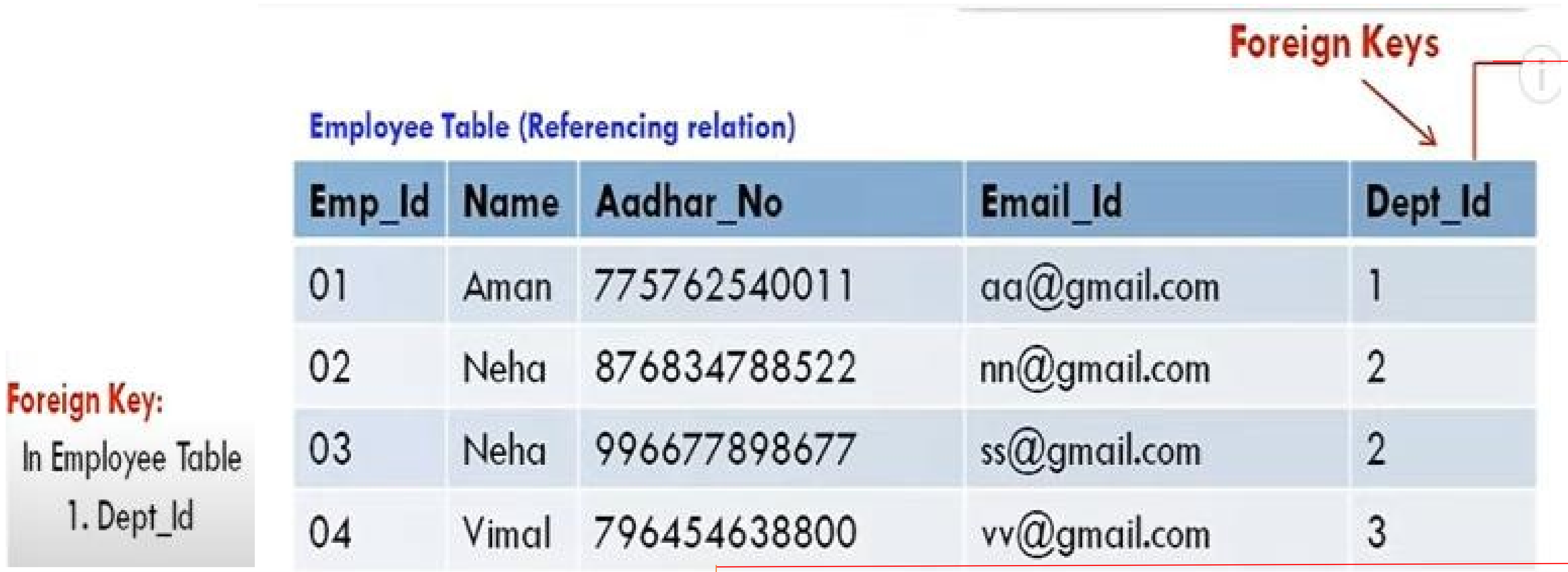
➢ Foreign key may have a name other than that of primary key.

➢ Foreign key can take the NULL values

The **purpose** of the **foreign key** is

➢ To ensure (or maintain) **referential integrity** of the data.

Eg: Foreign Key



Foreign Keys

Employee Table (Referencing relation)

Emp_Id	Name	Aadhar_No	Email_Id	Dept_Id
01	Aman	775762540011	aa@gmail.com	1
02	Neha	876834788522	nn@gmail.com	2
03	Neha	996677898677	ss@gmail.com	2
04	Vimal	796454638800	vv@gmail.com	3

Foreign Key:
In Employee Table
1. Dept_Id

Department Table (Referenced relation)

Primary Key

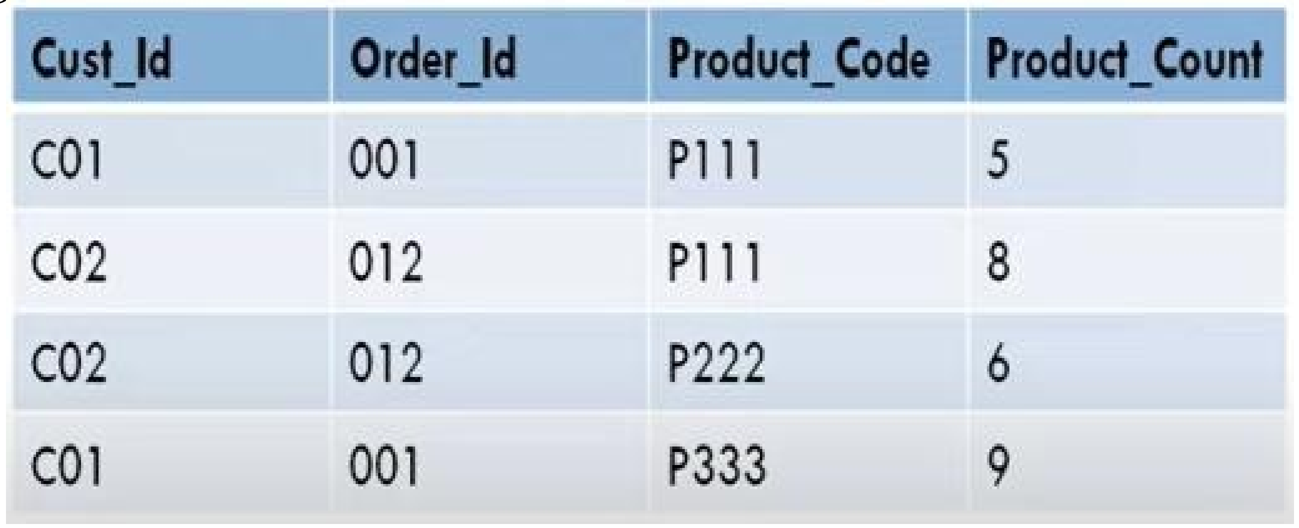
Dept_id	Dept_name
1	Sales
2	Marketing
3	HR

6. Composite Key

➢ A key that has more than one attributes is known as composite key. It is also known as compound key.

Composite key:

➢ {Cust_Id , Product_Code}



Cust_Id	Order_Id	Product_Code	Product_Count
C01	001	P111	5
C02	012	P111	8
C02	012	P222	6
C01	001	P333	9

Extended Entity Relationship (ER) Features

- As the complexity of data increased in the late 1980s, it became more & more difficult to use the traditional ER Model for database modelling. Hence some improvements or enhancements were made to the existing ER model to make it able to handle the complex applications better.
- Hence, as part of the Extended ER Model ,along with other improvements, three new concepts were added to the existing ER Model:

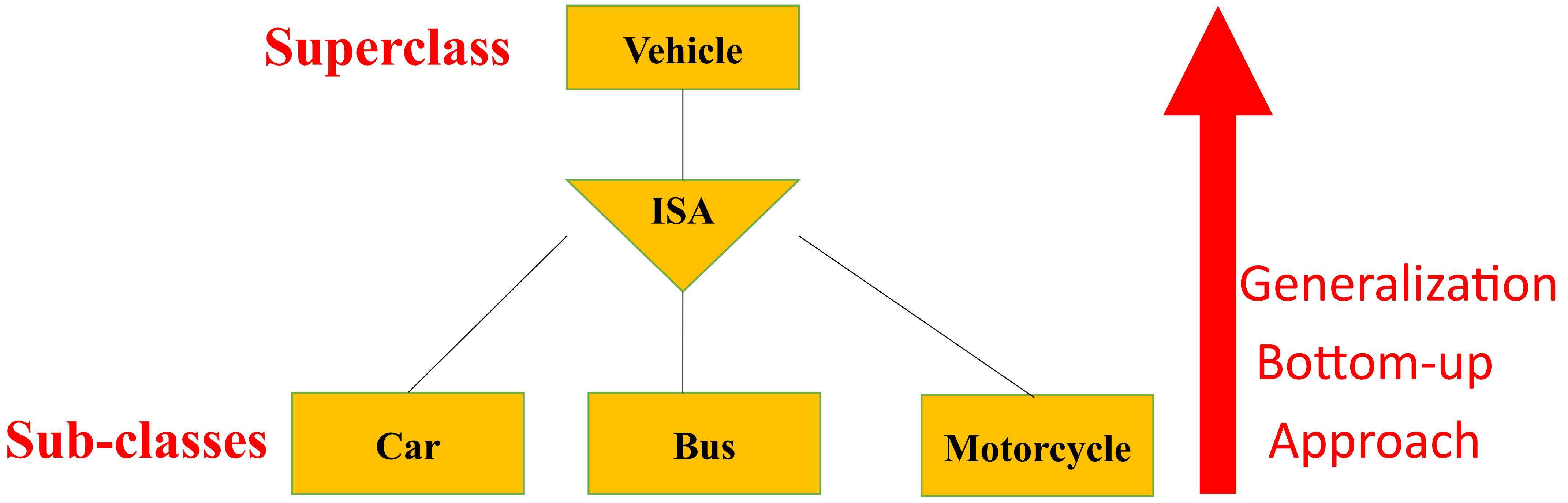
1. Generalization
2. Specialization
3. Aggregation

Generalization

- ➢Generalization is the process of extracting common properties from a set of entities & create a generalized entity from it.
- ➢Generalization is A “BOTTOM-UP APPROACH” in which two or more entities can be combined to form a higher level entity if they have some attributes in common
- Subclasses are combined to make a superclass
- ➢Generalization is used to emphasize the similarities among lower-level entity set & to hide differences in the schema

Eg: Generalization

- Consider we have 3 sub entities Car, Bus & Motorcycle.Now these three entities can be generalized into one higher-level entity (or superclass) named as Vehicle

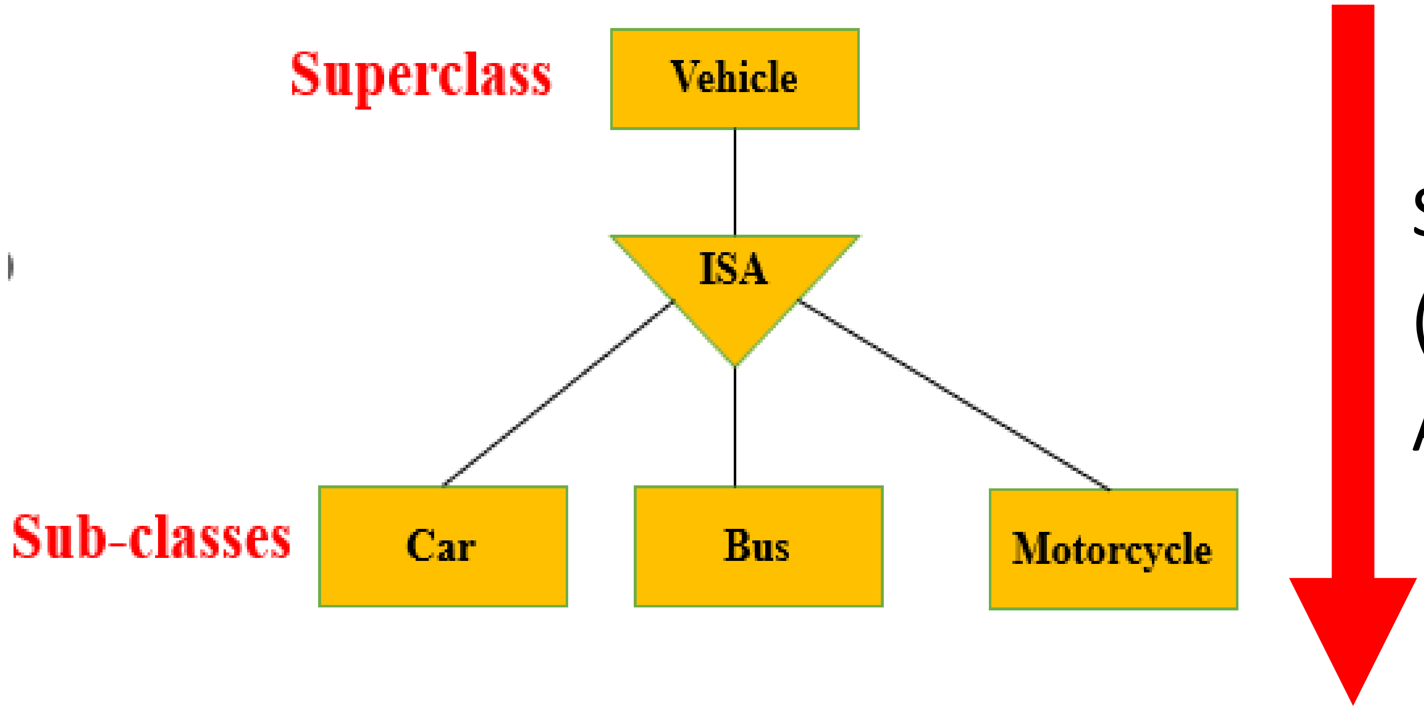


Specialization

- Specialization is **opposite** of Generalization.
- In Specialization, an entity is broken down into sub-entities based on their characteristics.
- Specialization is a "**Top-down approach**" where higher level entity is specialized into two or more lower level entities.
- Specialization is used to identify the subset of an entity set that shares some distinguishing characteristics.
- Specialization can be repeatedly applied to refine the design of schema .
- It is depicted by triangle component labelled as **ISA**.

Eg: Specialization

- Vehicle entity can be a Car, Truck or Motorcycle.
- Normally, the superclass is defined first, the subclass and its related attributes are defined next, & relationship set are then added



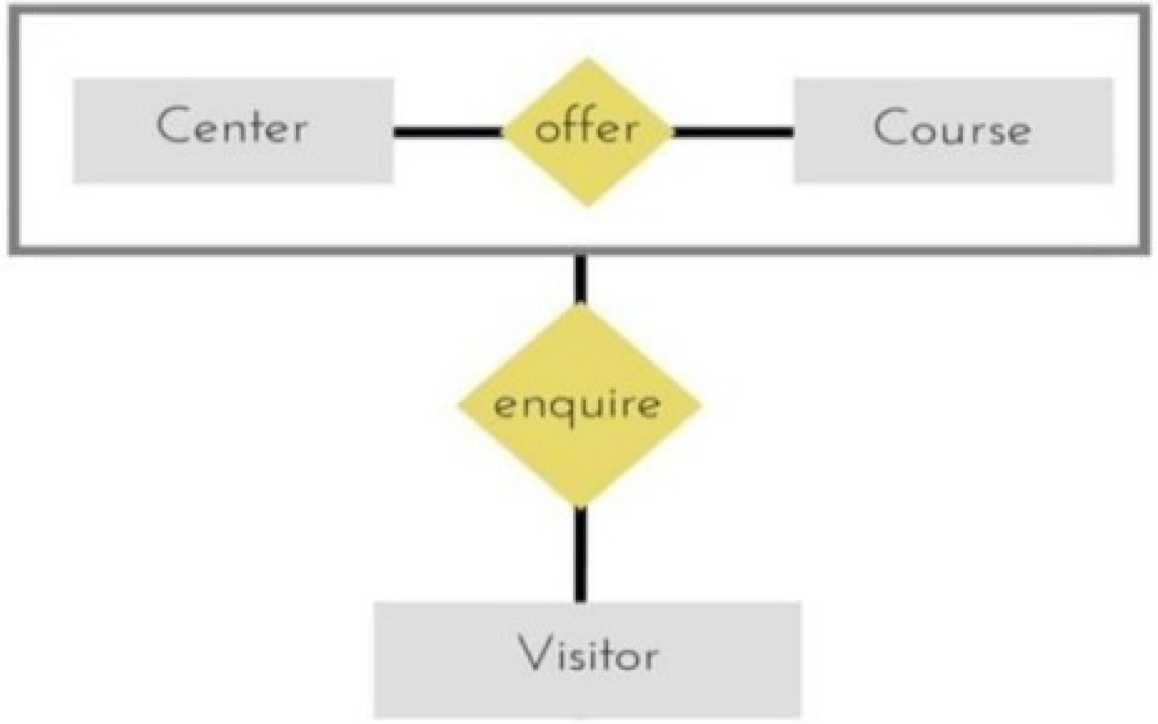
Specialization (Top-down Approach)

Aggregation

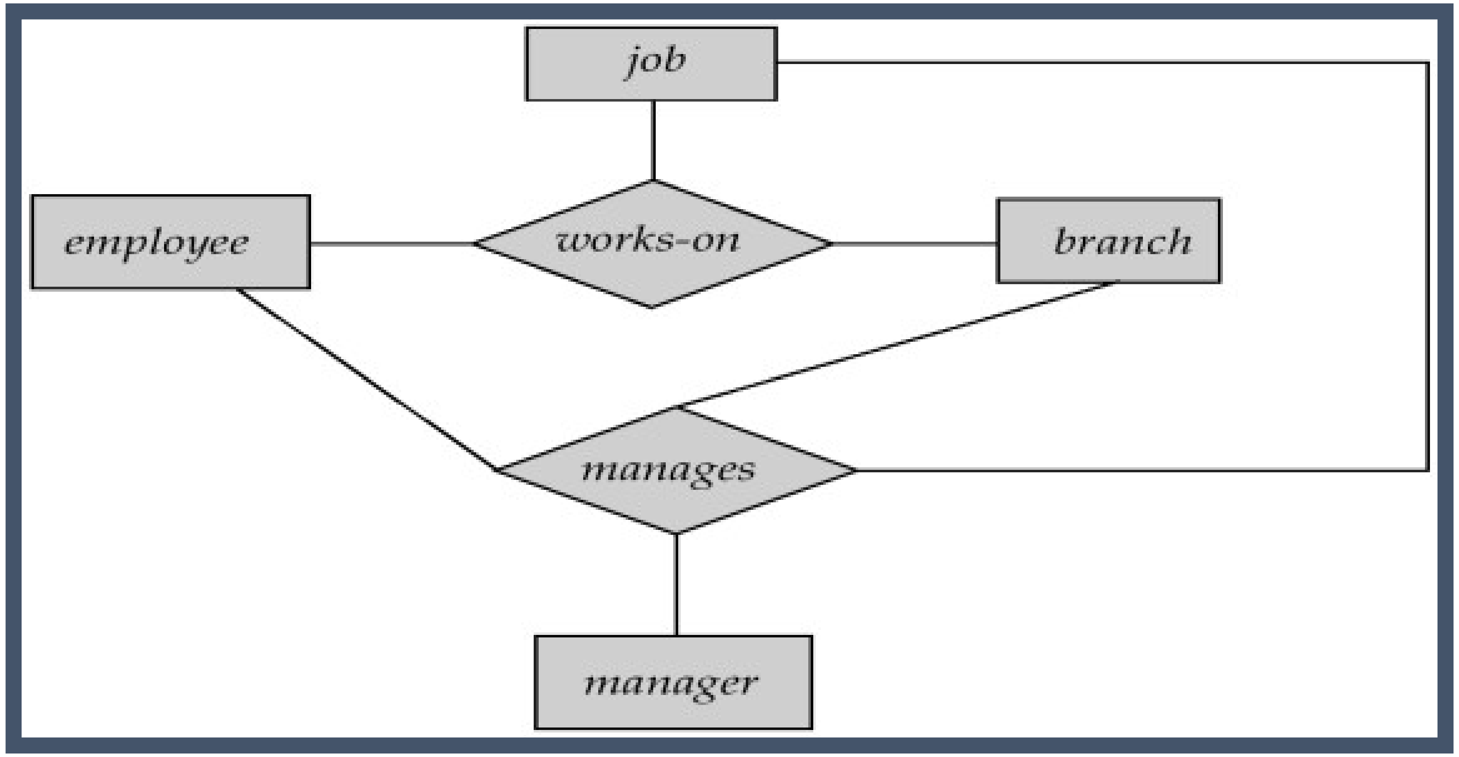
- **Aggregation** is used when we need to express a **relationship among** relationships
- Aggregation is an **abstraction** through which **relationships** are treated as **higher level entities**.
- Aggregation is a process when a relationship between two entities is considered as a **single entity** & again this single entity has a relationship with another entity.

Eg: Aggregation

Here the relation between Center and Course, is acting as an Entity in relation with Visitor.



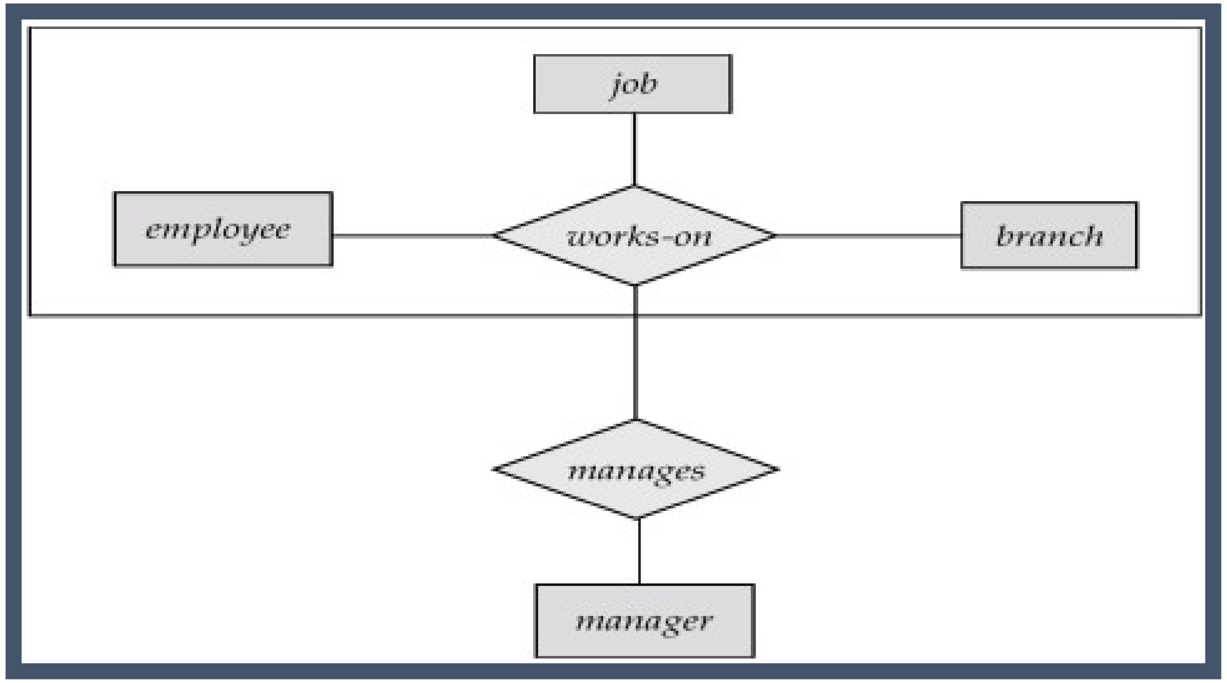
Eg (Aggregation):
Consider the ternary relationship works-on, which we saw earlier.
Suppose we want to record managers for tasks performed by an employee at a branch



- Relationship sets *works-on* and *manages* represent overlapping information
  - Every *manages* relationship corresponds to a *works-on* relationship
  - However, some *works-on* relationships may not correspond to any *manages* relationships
    - So we can’t discard the *works-on* relationship
- Eliminate this redundancy via *aggregation* (*objectification*)
  - Treat relationship as an abstract entity
  - Allows relationships between relationships
  - Abstraction of relationship into new entity
- Without introducing redundancy, the following diagram represents:
  - An employee works on a particular job at a particular branch
  - An employee, branch, job combination may have an associated manager

E-R Diagram With Aggregation

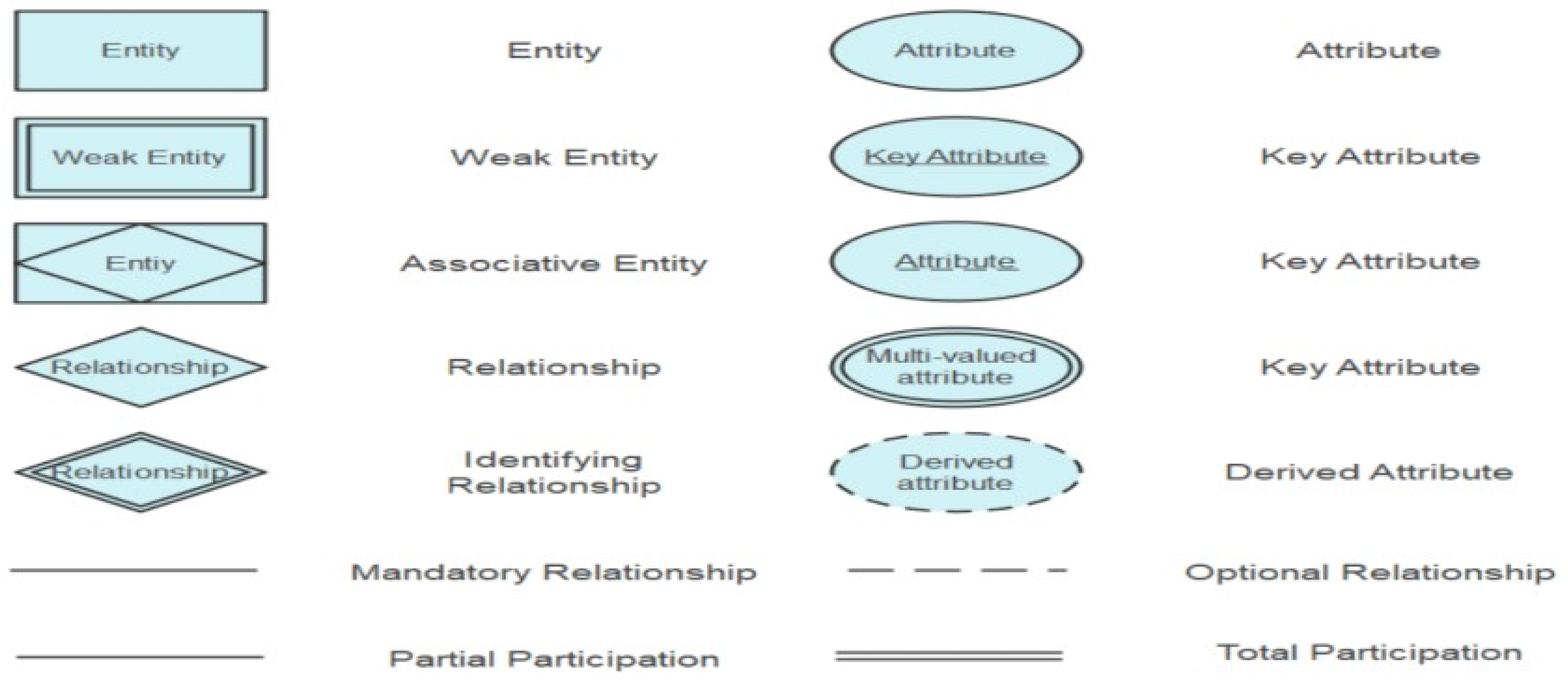
- Without introducing redundancy, the following diagram represents:
  - An employee works on a particular job at a particular branch
  - An employee, branch, job combination may have an associated manager

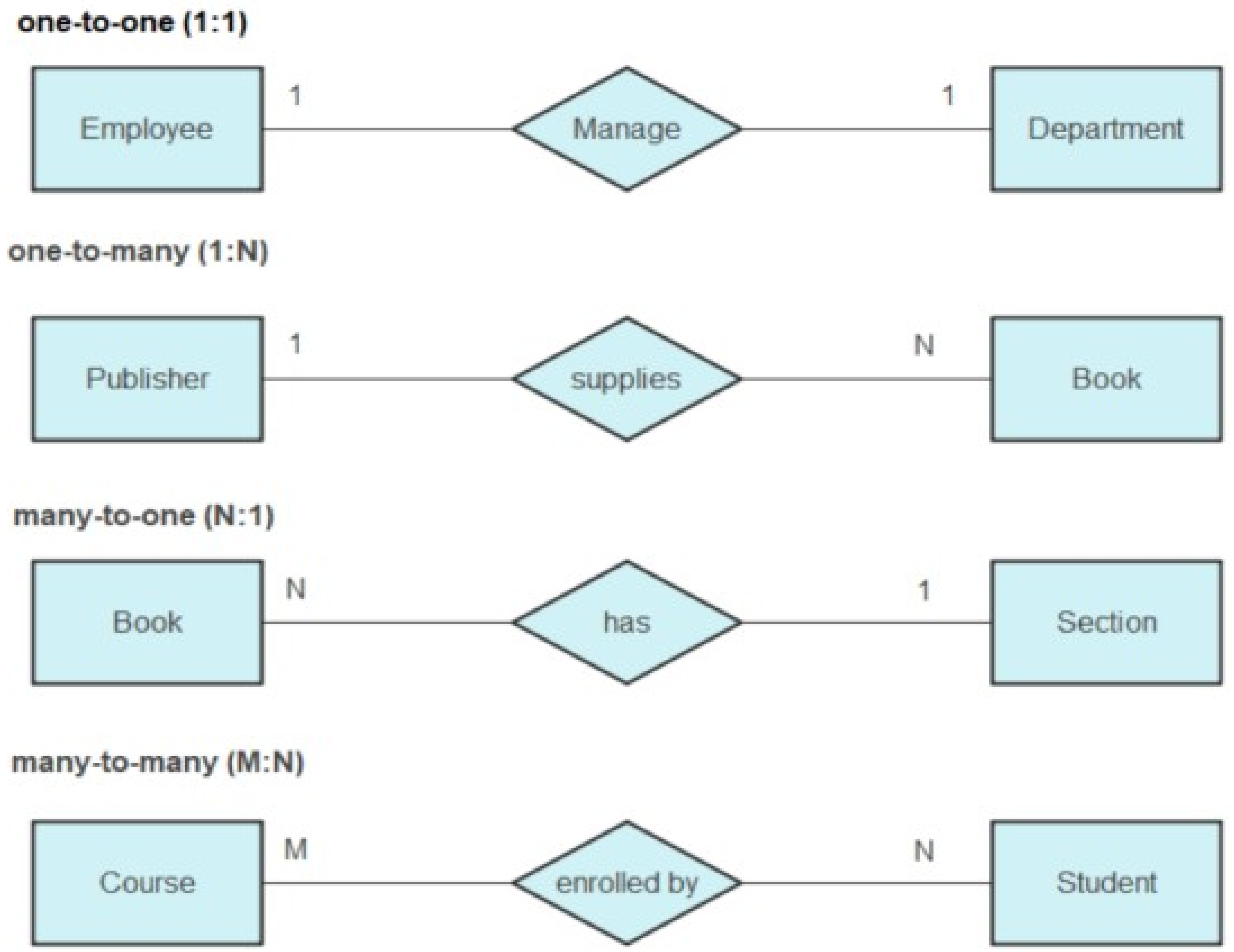


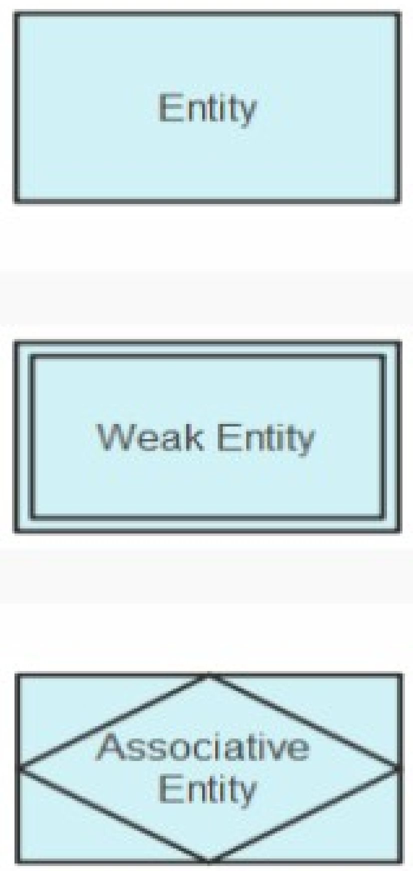
Steps to draw an ER Diagram

1. Identify the entities
2. Identify the attributes
3. Identify the primary key
4. Identify the relationships and participation constraints

Symbols used in E-R Notation




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
Weak Entity
Associative Entity

Banking Databse E-R model

1. **Identify the entities**

i. Branch

ii. Customer

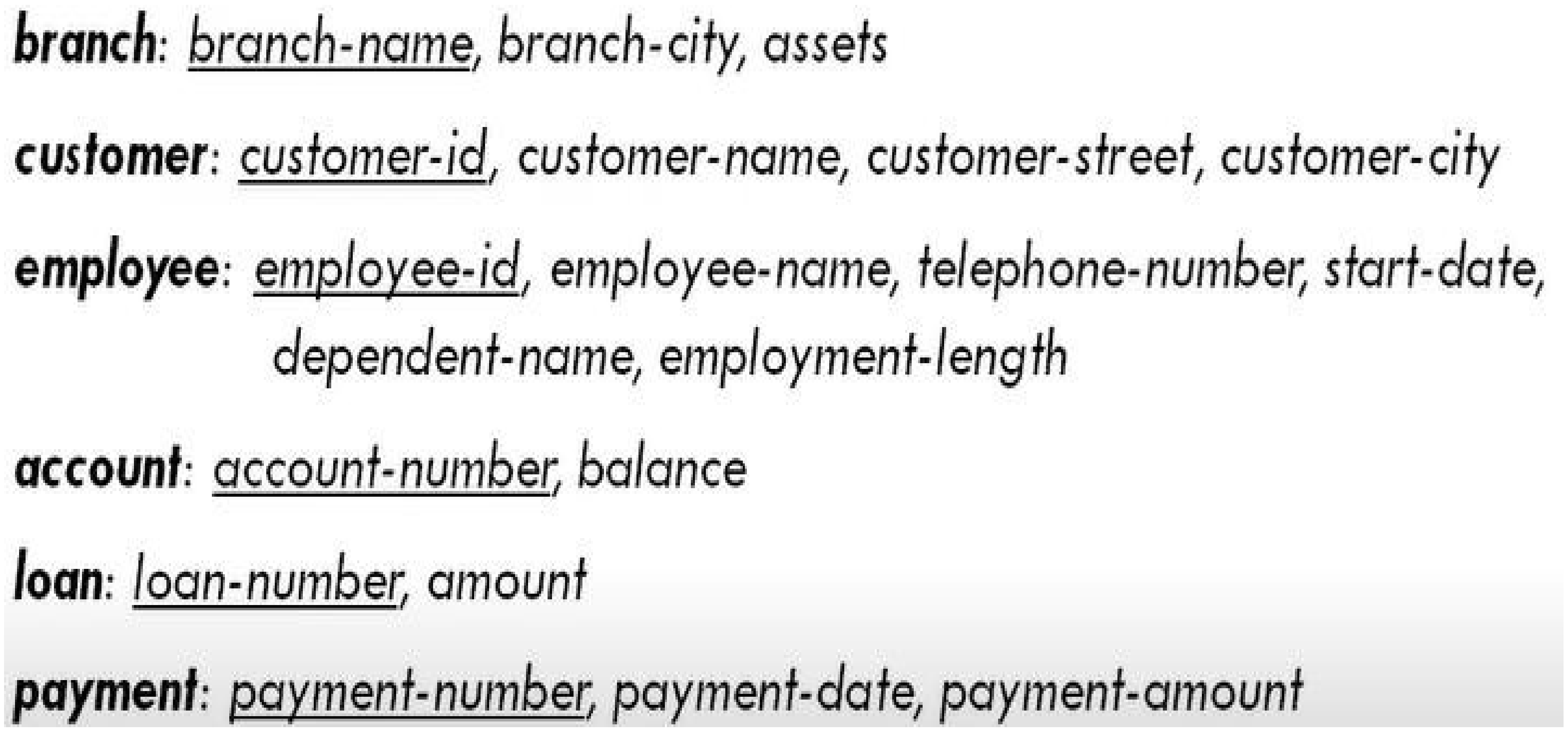
iii. Account

iv. Loan

v. Employee

vi. Payment

2. IDENTIFY THE ATTRIBUTES



branch: *branch-name, branch-city, assets*

customer: *customer-id, customer-name, customer-street, customer-city*

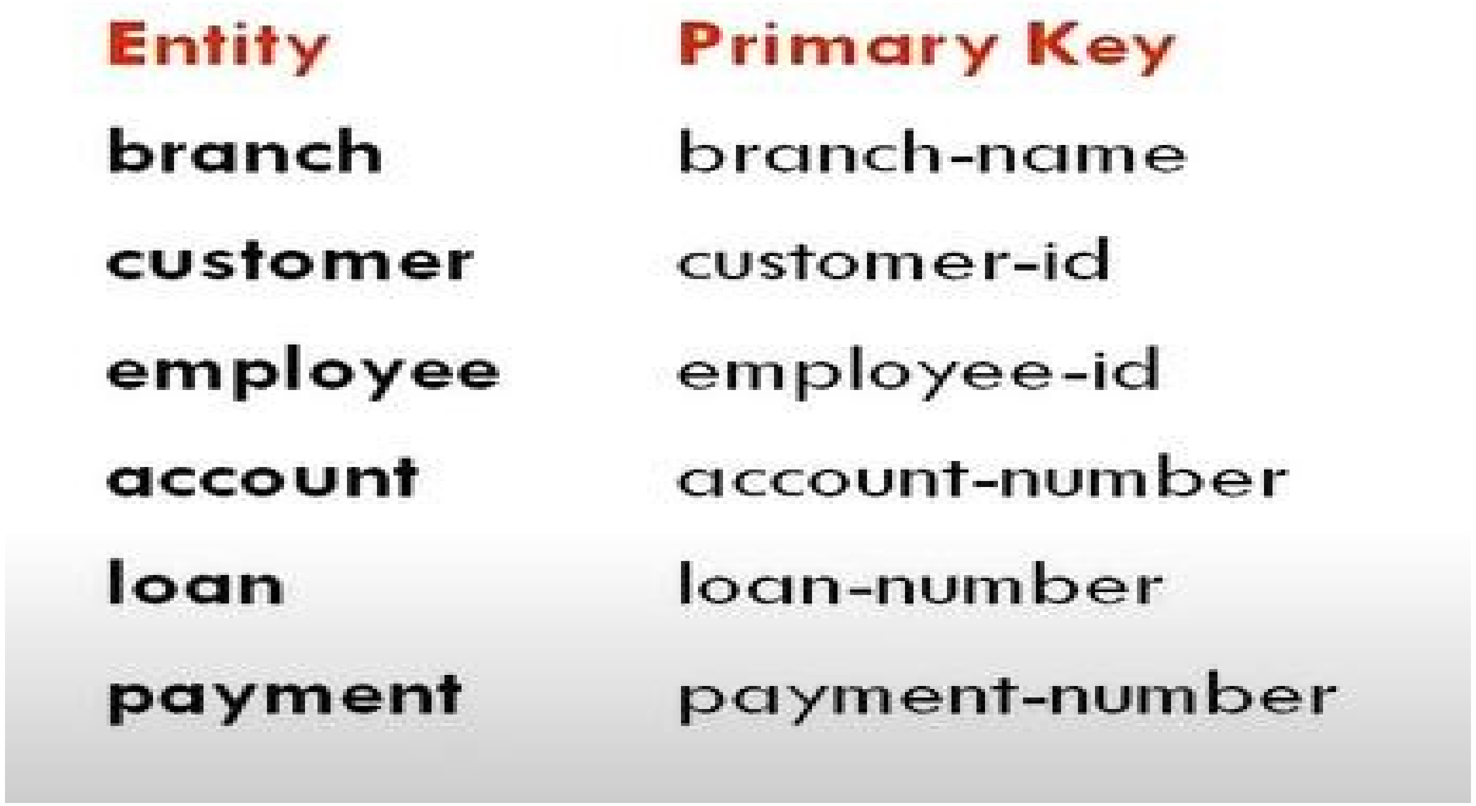
employee: *employee-id, employee-name, telephone-number, start-date, dependent-name, employment-length*

account: *account-number, balance*

loan: *loan-number, amount*

payment: *payment-number, payment-date, payment-amount*

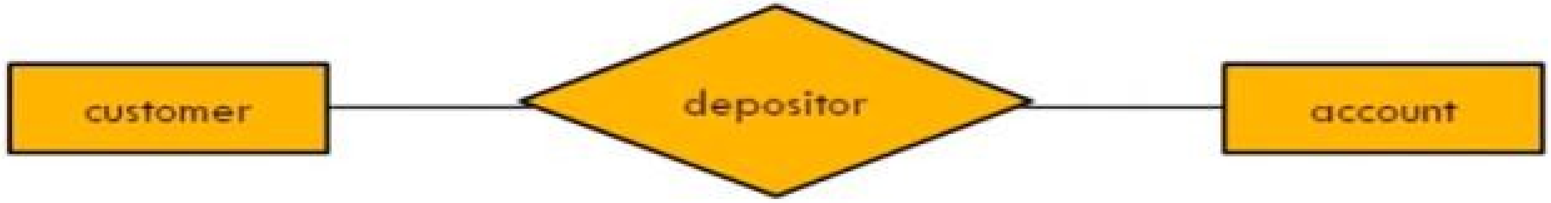
3. Identify the Primary Key



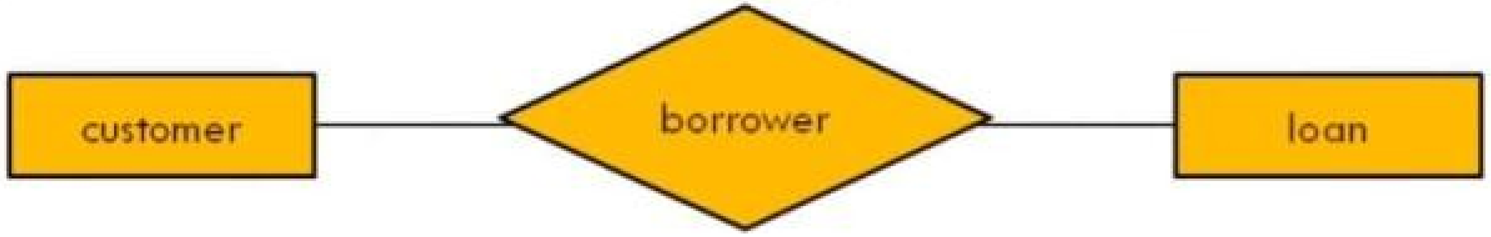
Entity	Primary Key
branch	branch-name
customer	customer-id
employee	employee-id
account	account-number
loan	loan-number
payment	payment-number

4. Identify the Relationships & Participation Constraints

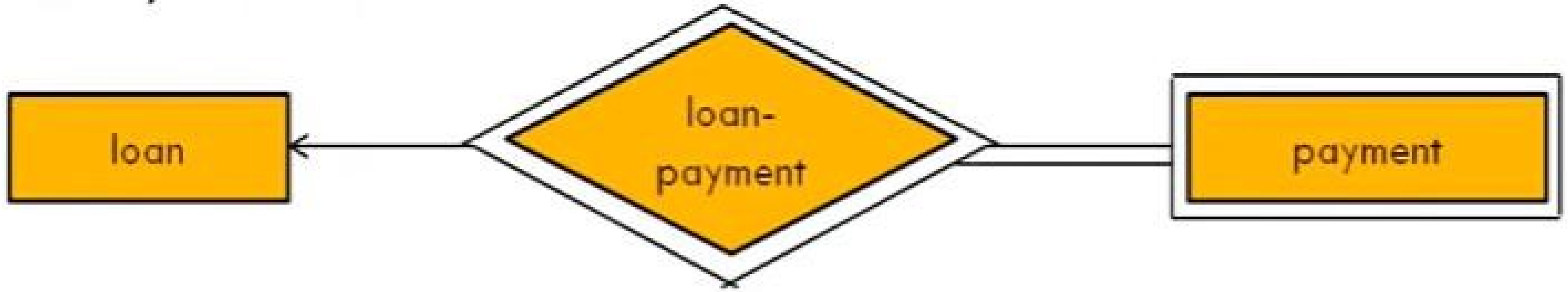
A **customer** can have multiple **accounts** and an account can be held by multiple customers. Cardinality will be M : N



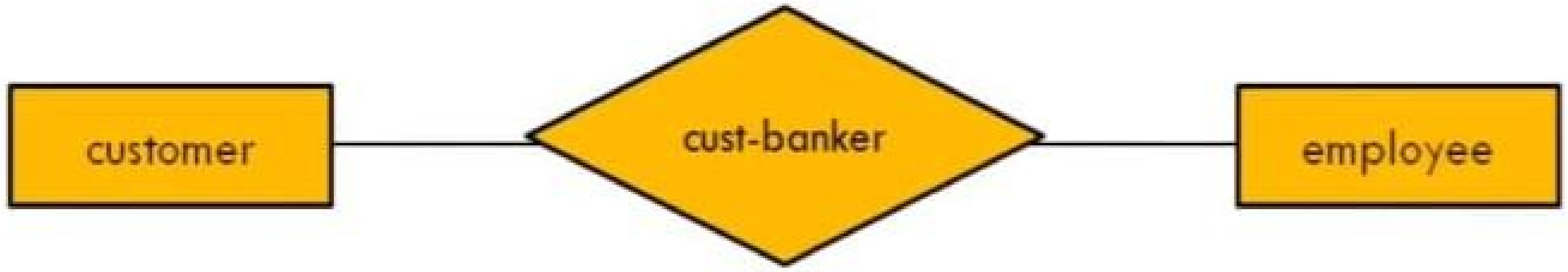
Customers are allowed to take **loans** from the bank, a loan can be held by one or more customers. Cardinality will be M : N



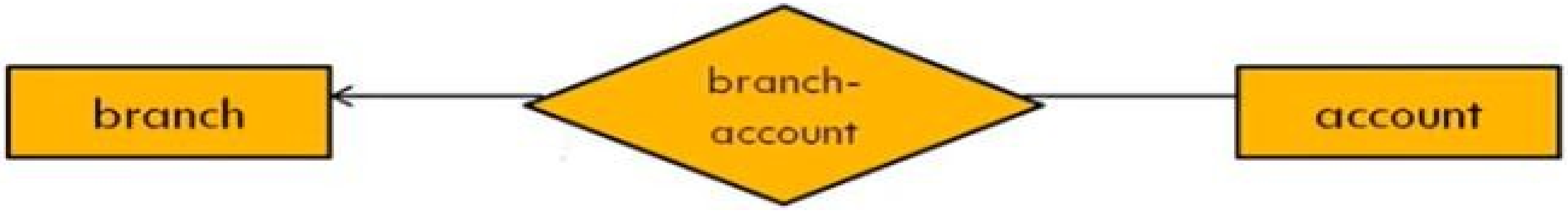
Each **loan** is paid back with the multiple **payments** having payment number. Cardinality will be 1 : N



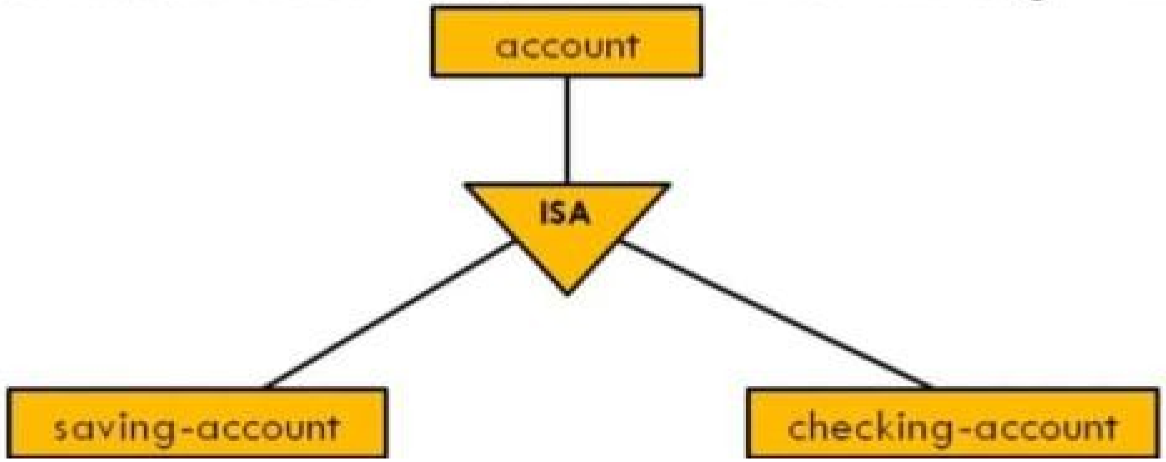
A **customer** can be an **employee** of the bank. Cardinality will be M : N



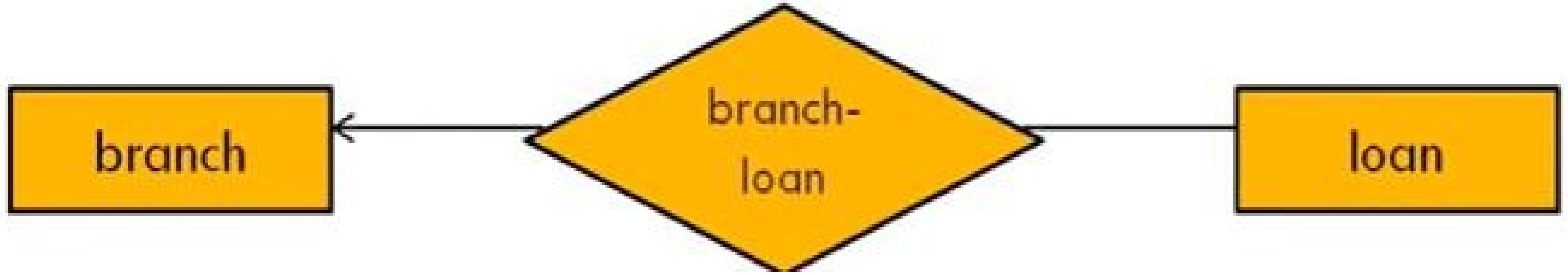
A **branch** can have multiple **accounts.** Cardinality will be 1 : N



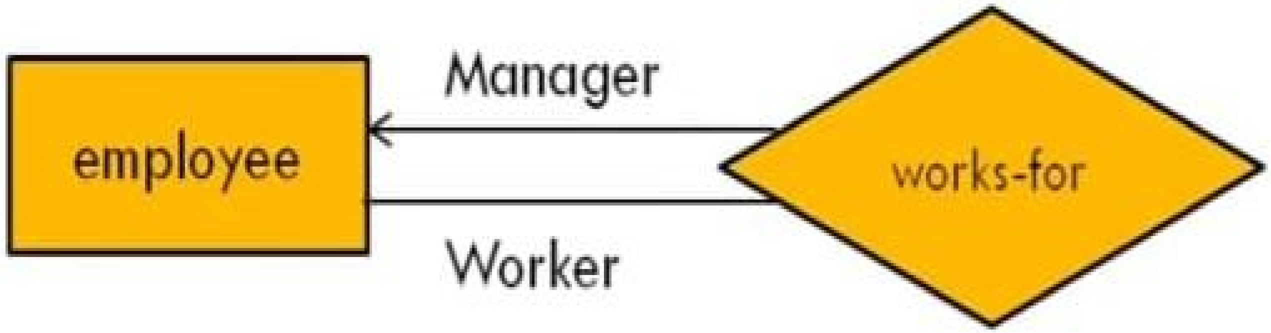
An **account** can be divided into two accounts: saving account and checking account

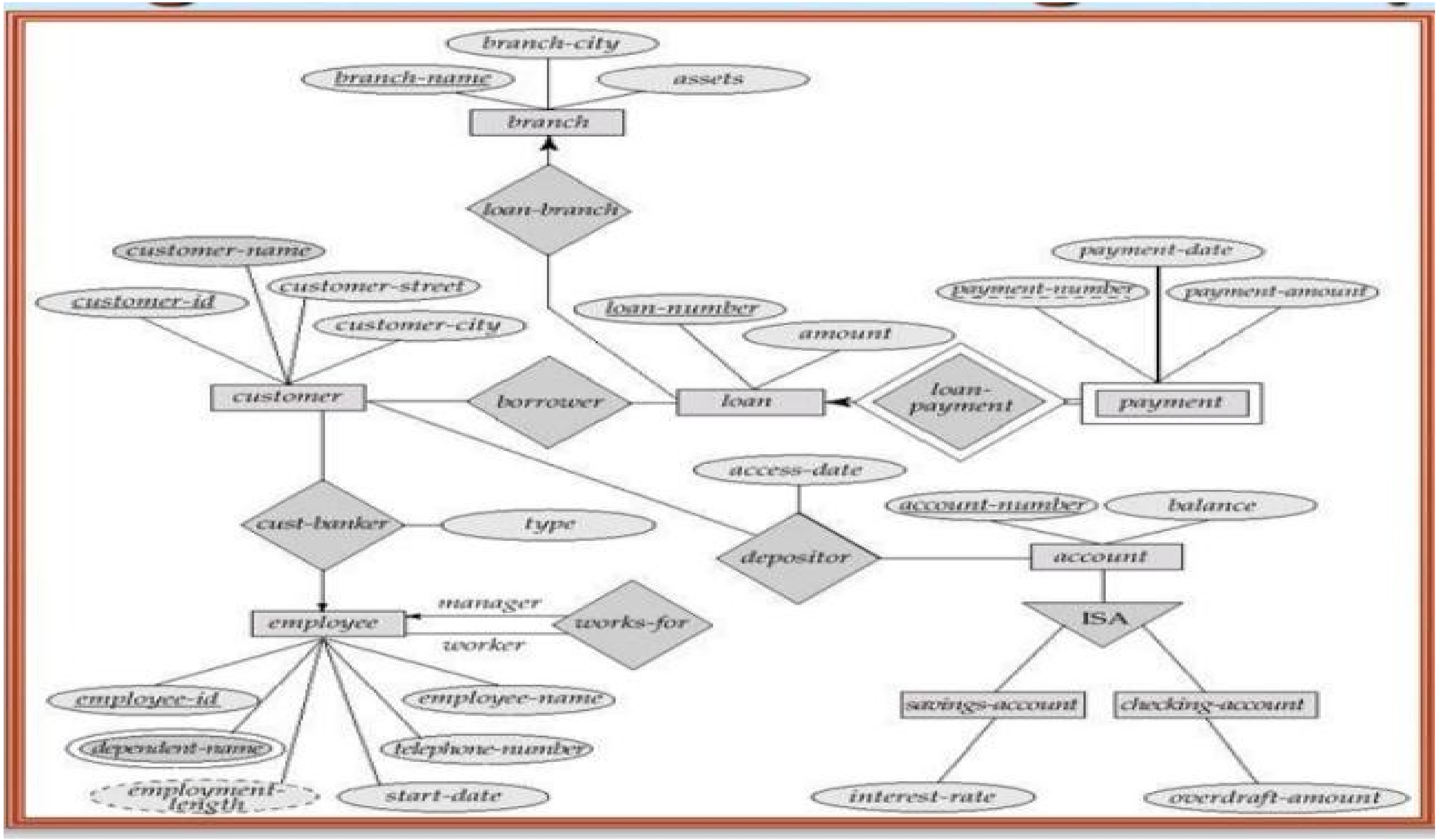


A **branch** of bank give **loans** to the customers. Cardinality will be 1 : N



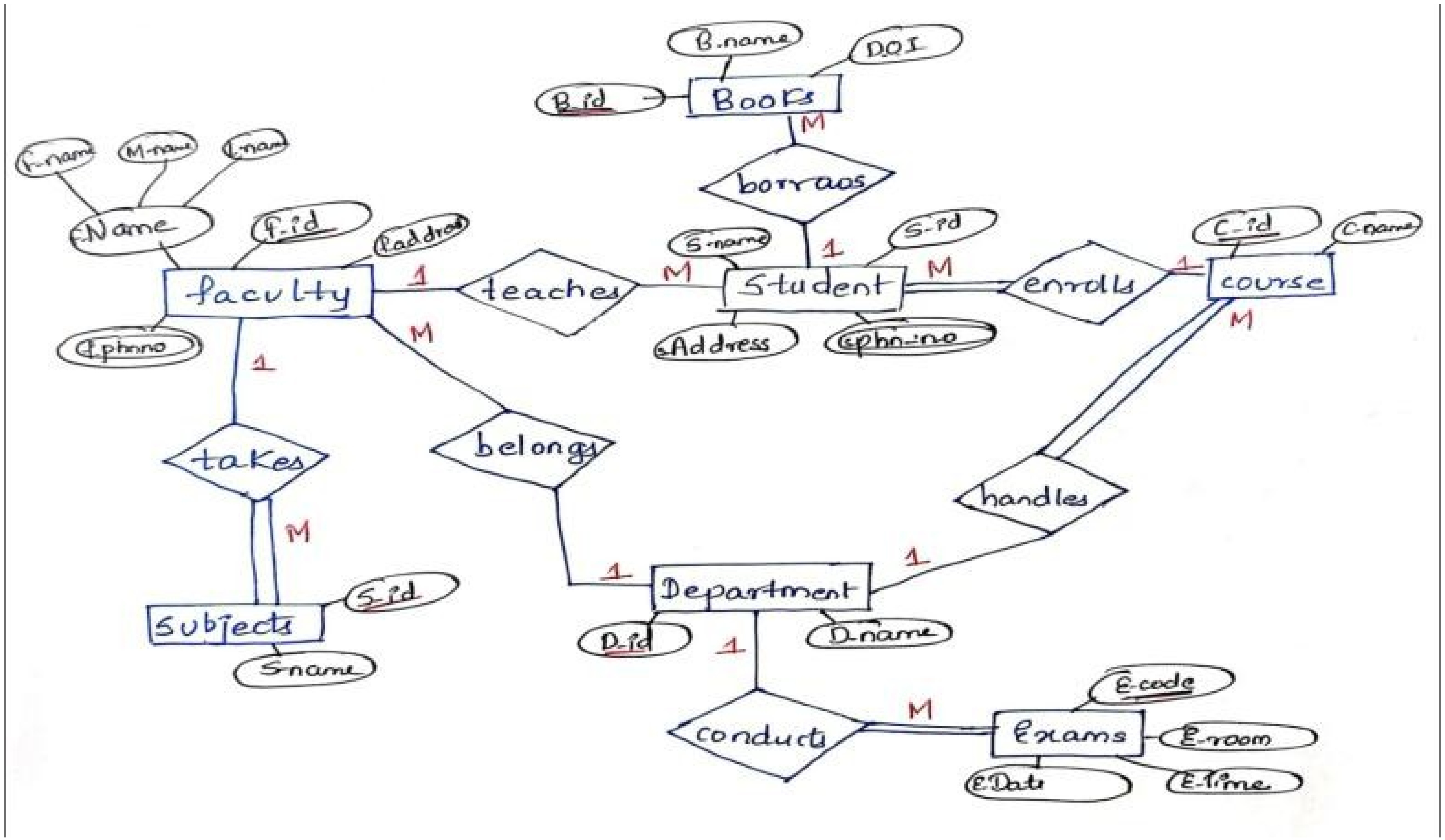
Employees can be a **manger** or a **worker** of that particular branch. (Recursive Relationship exist)




branch-city
branch-name
assets
branch
loan-branch
customer-name
customer-street
customer-id
customer-city
loan-number
amount
payment-date
payment-number
payment-amount
customer
borrower
loan
loan-payment
payment
access-date
account-number
balance
cust-banker
type
depositor
account
manager
works-for
worker
employee
ISA
employee-id
employee-name
dependent-name
telephone-number
employment-length
start-date
savings-account
checking-account
interest-rate
overdraft-amount

ER-Diagram on College Management System

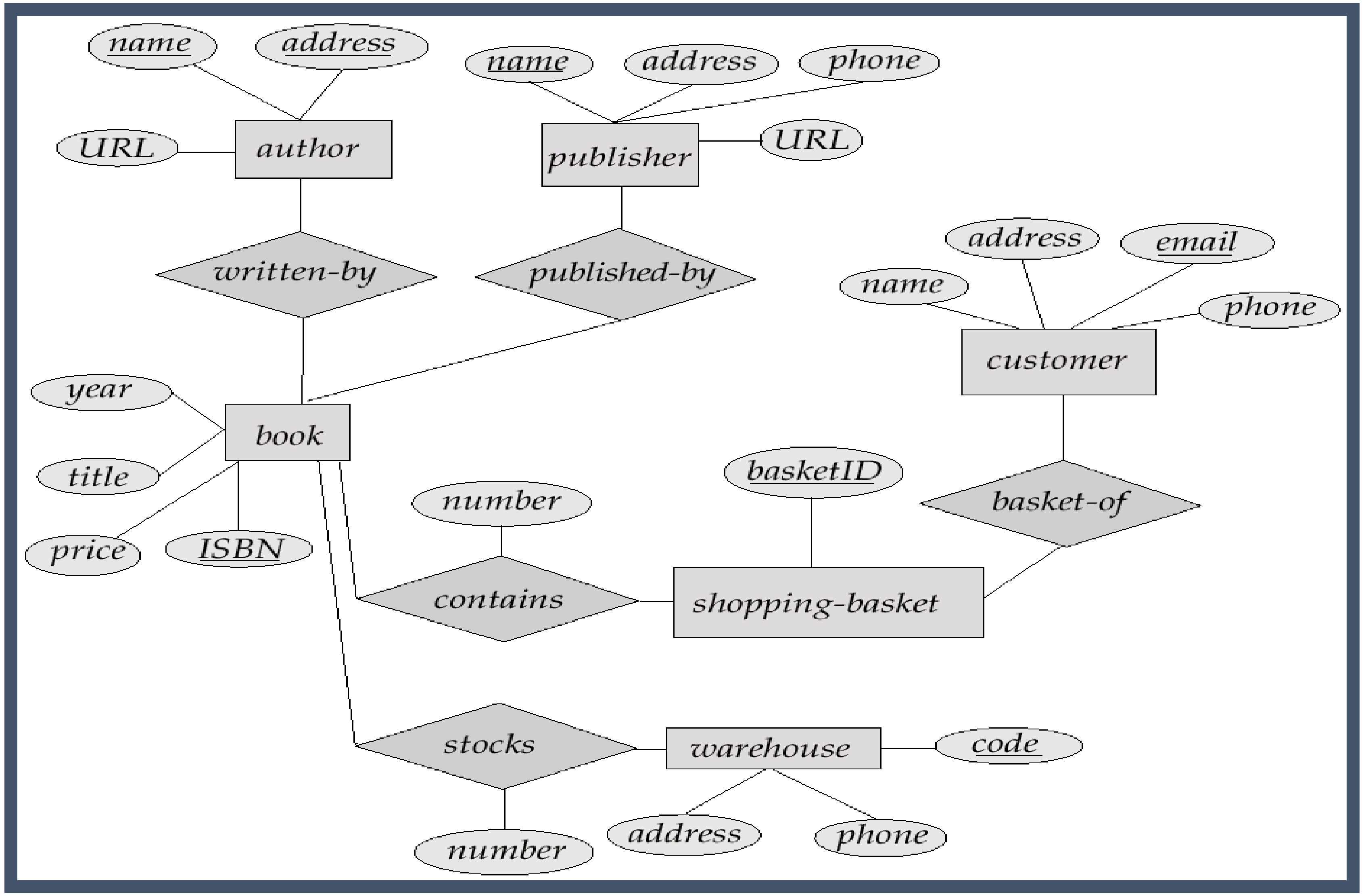
-Presented by Charul Singh



-Presented by Charul Singh

ER-Diagram on Online Book Shopping Cart

-Presented by Charul Singh


name
address
author
URL
name
address
phone
publisher
URL
written-by
published-by
address
email
name
phone
customer
year
book
title
price
ISBN
number
basketID
basket-of
contains
shopping-basket
stocks
warehouse
code
number
address
phone