

Database Design & Applications

Entity Relationship Diagrams

Software/Systems

Software/Systems development is a systematic process which includes phases:

- Planning
- Analysis
- Design
- Testing
- Deployment
- Maintenance

Software Development

- System Analysis phase is systematic approach, which uses graphical tools that analyze and refine the objectives of an existing system and develop a new system specification which can be easily understandable by user
- Various graphical tools are used to represent the two main requirements of the system:
 - Functional
 - Data
- Entity Relationship Diagram is the tool to model the data requirements of a system.

Entity Relationship Diagram

A Data Model is usually the result of systematic analysis to define and describe the data that must be captured, stored and retrieved to support the business

The highest level of abstraction for the data model is called Entity Relationship Diagram (ERD).

ERD is a graphical representation of data requirements for a database

Entity Relationship Diagram

There are three components of ERD:

Entities

Attributes

Relationships

Entity

- Entities are the people, places, things, events and concepts of interest to an organisation.
- Entities are represented Rectangle

Book

Project

Publisher

Job

Company

File

Attribute

Properties or characteristics possessed by an entity are called as attributes

Represented as ellipse

Attributes of EMPLOYEE entity set :

- Name
- Age
- Employee_id
- Salary

Domain

The set of possible values for an attribute is called the domain of the attribute.

The domain of attribute marital status is just the four values: single, married, divorced, widowed.

The domain of the attribute month is the twelve values ranging from January to December.

Simple v/s Composite Attribute

Simple Attribute

An attribute that cannot be divided into simpler components

- age of an employee
- pub-id of a book
- title of a book.

Composite Attribute

An attribute that can be split into components

- Date of birth of the student can be split into:
 - Day
 - Month
 - Year

Single v/s Multi-valued Attribute

Single valued

Attributes that can take on only a single value for each entity instance.

- Age of employee
- publisher_id of a publisher

Multi-valued

Attributes that can take many values for each entity instance.

- Skill set of employee
- Hobbies of a Student

Stored v/s Derived Attribute

Stored Attribute

Attribute that need to be stored permanently

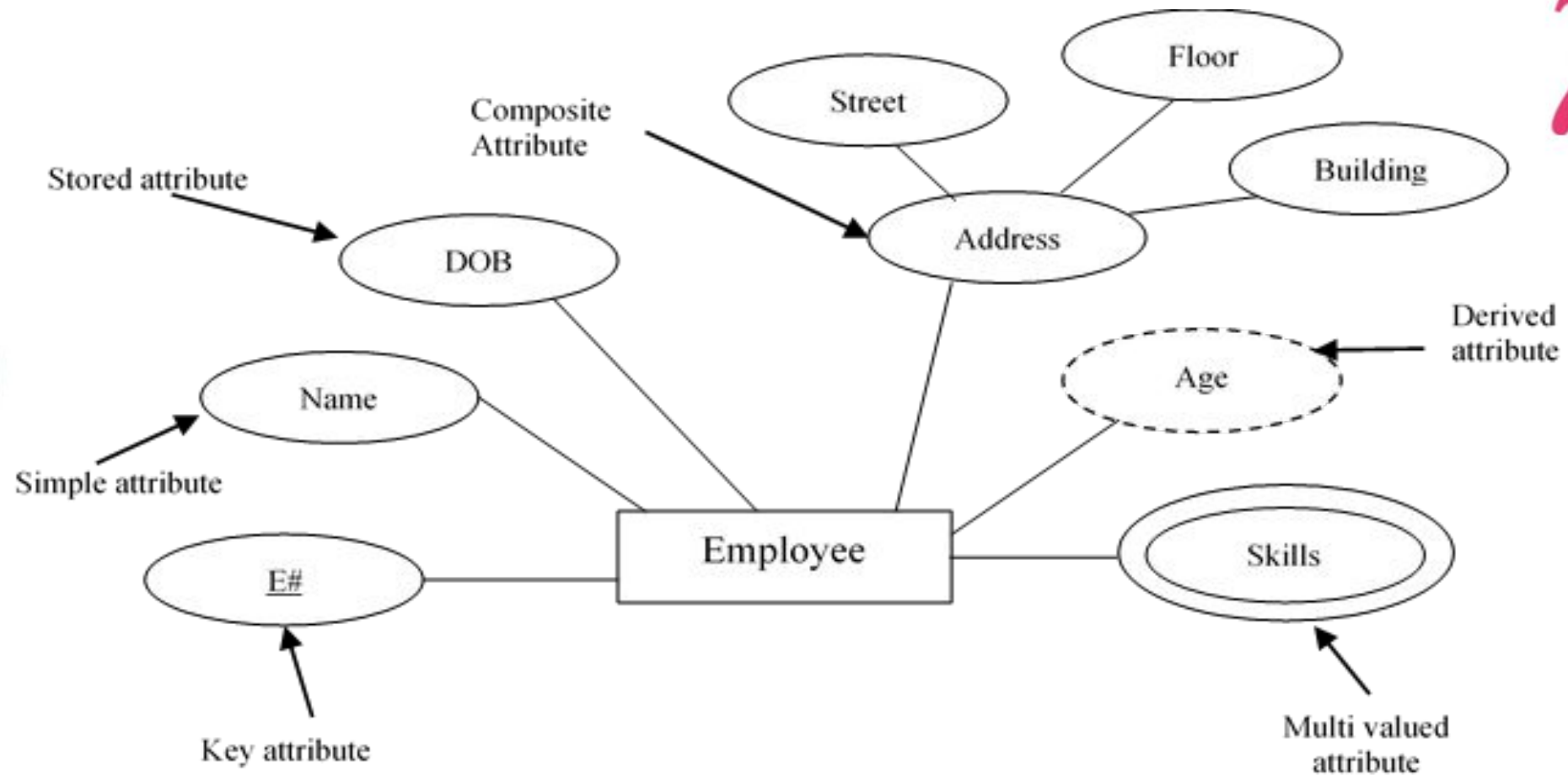
Name of an employee

Derived Attribute

Attribute that can be calculated based on other attributes

Years of service of employee can be calculated from date of joining and current date

Attributes



Relationship

Defines the set of all associations between two entity types.

Represented as Rhombus

Example:

- EMPLOYEEs work in a DEPARTMENT
- LAWYERs advise CLIENTs
- EQUIPMENT is allocated to PROJECTs
- TRUCK is a type of VEHICLE

Degree of a Relationship

The number of entities associated with the relationship.



One – Unary
employee - manager-of - employee

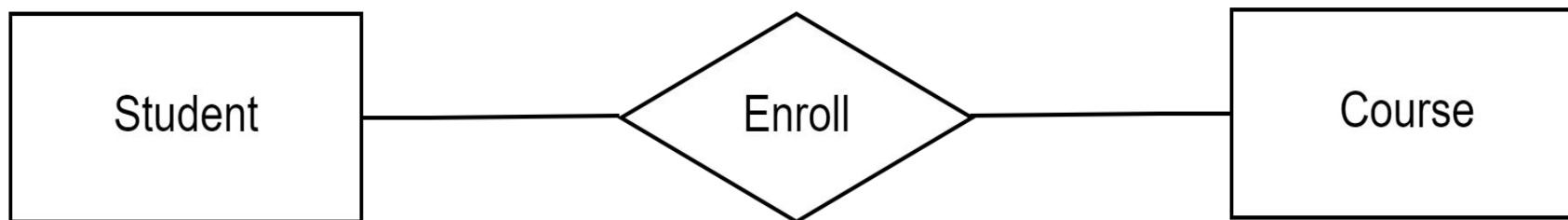


Two – Binary
employee - works-for – department

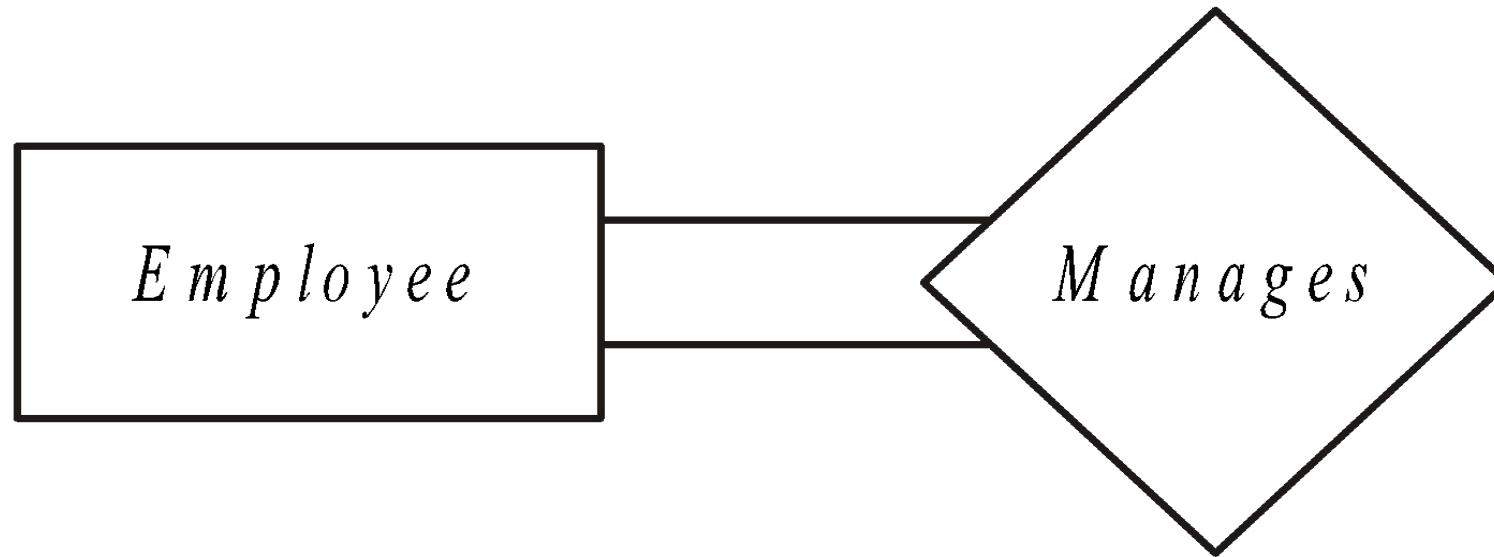


Three – Ternary
customer – purchase- item

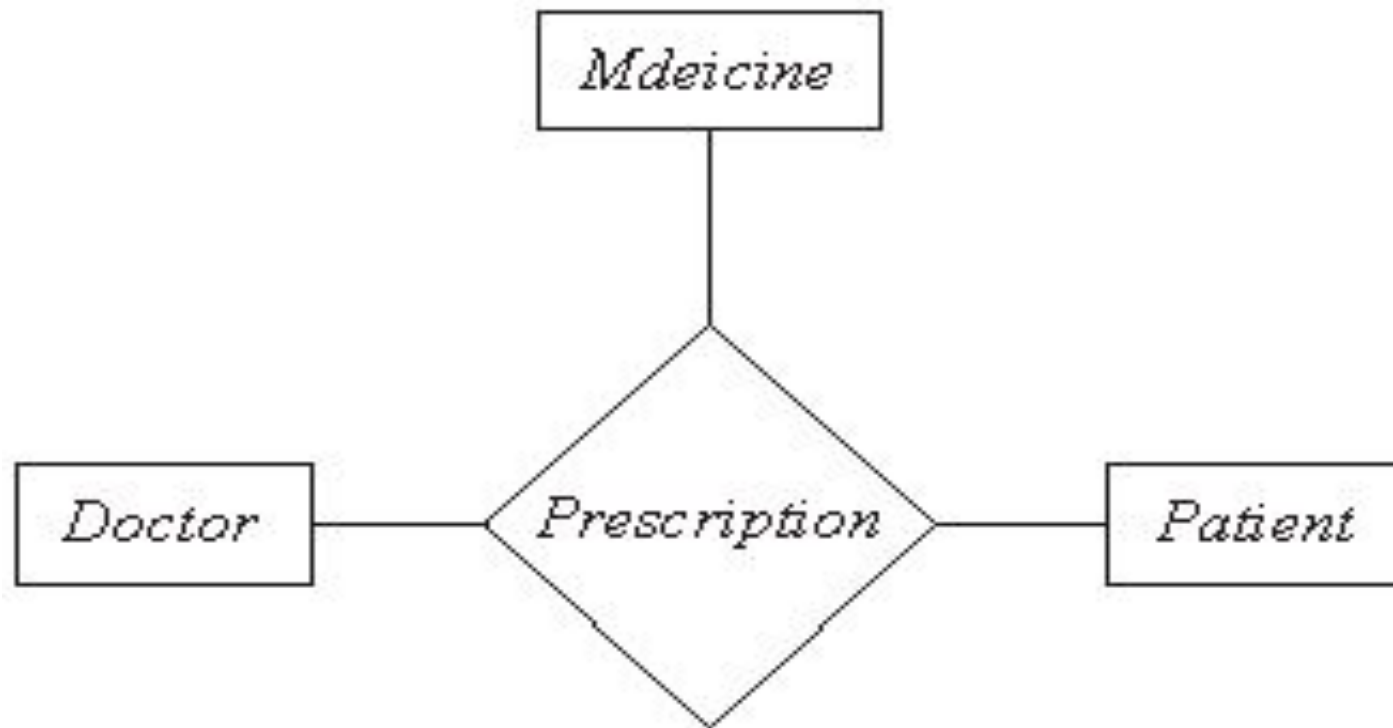
Binary Relationship



Unary Relationship



Ternary Relationship



Cardinality and Connectivity

- The cardinality of a relationship is the actual number of related occurrences for each of the two entities

one-to-one (1:1)

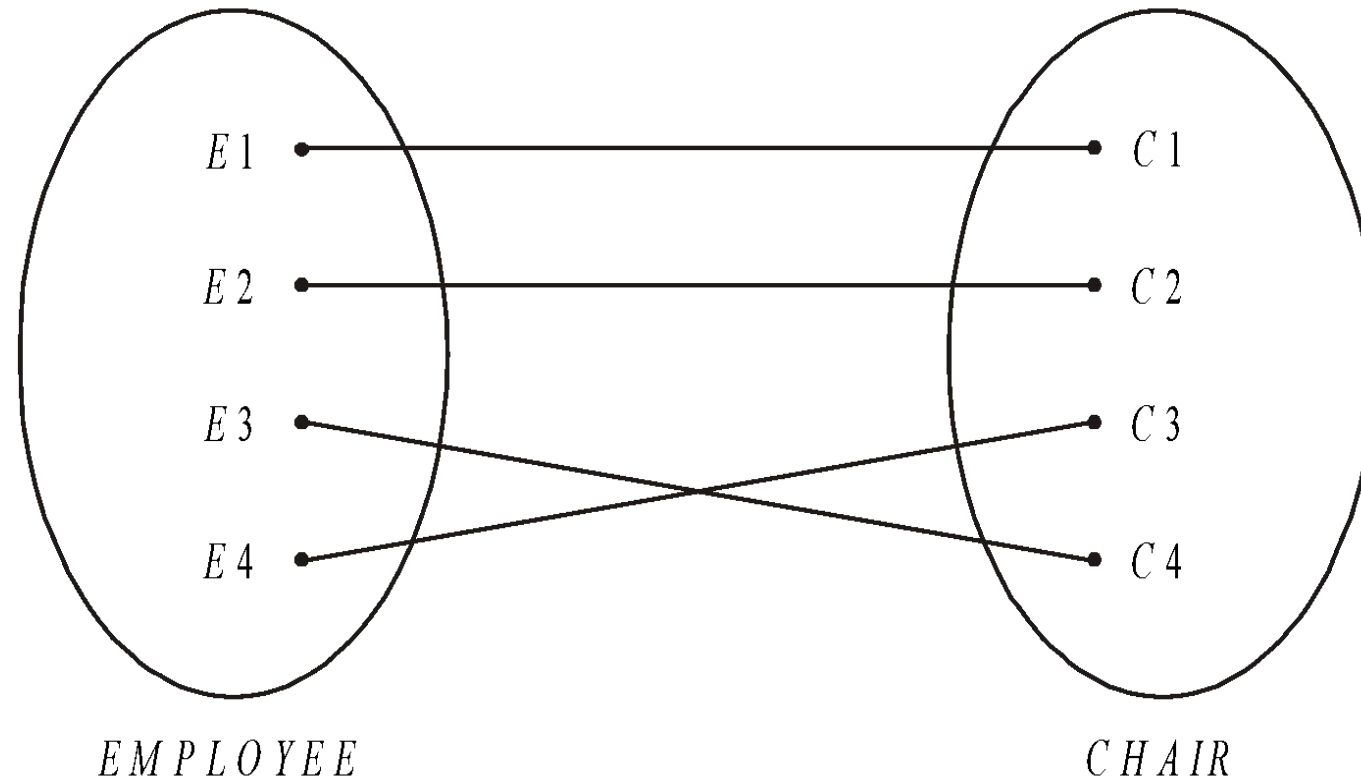
one-to-many (1:N)

many-to-one (M:1)

many-to-many (M:N)

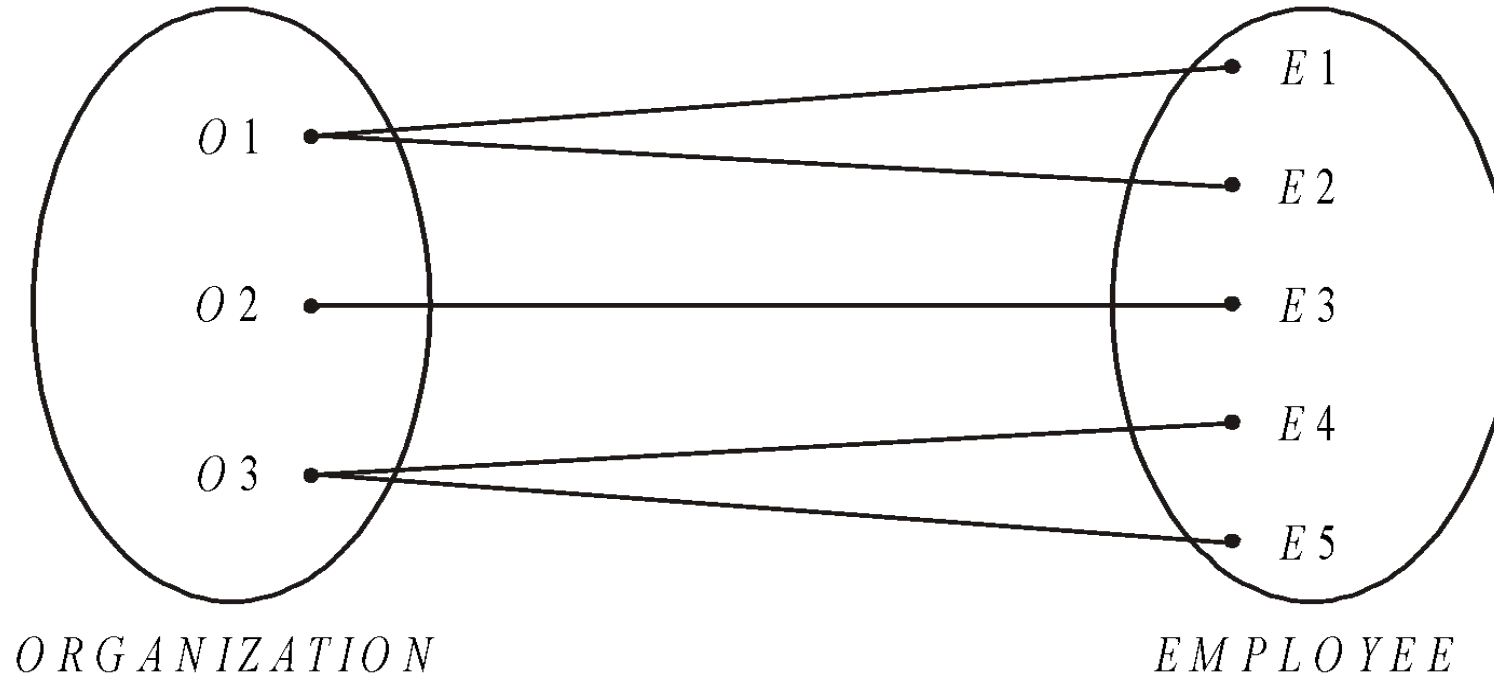
One-to-One (1:1)

When at most one instance of a entity A is associated with one instance of entity B



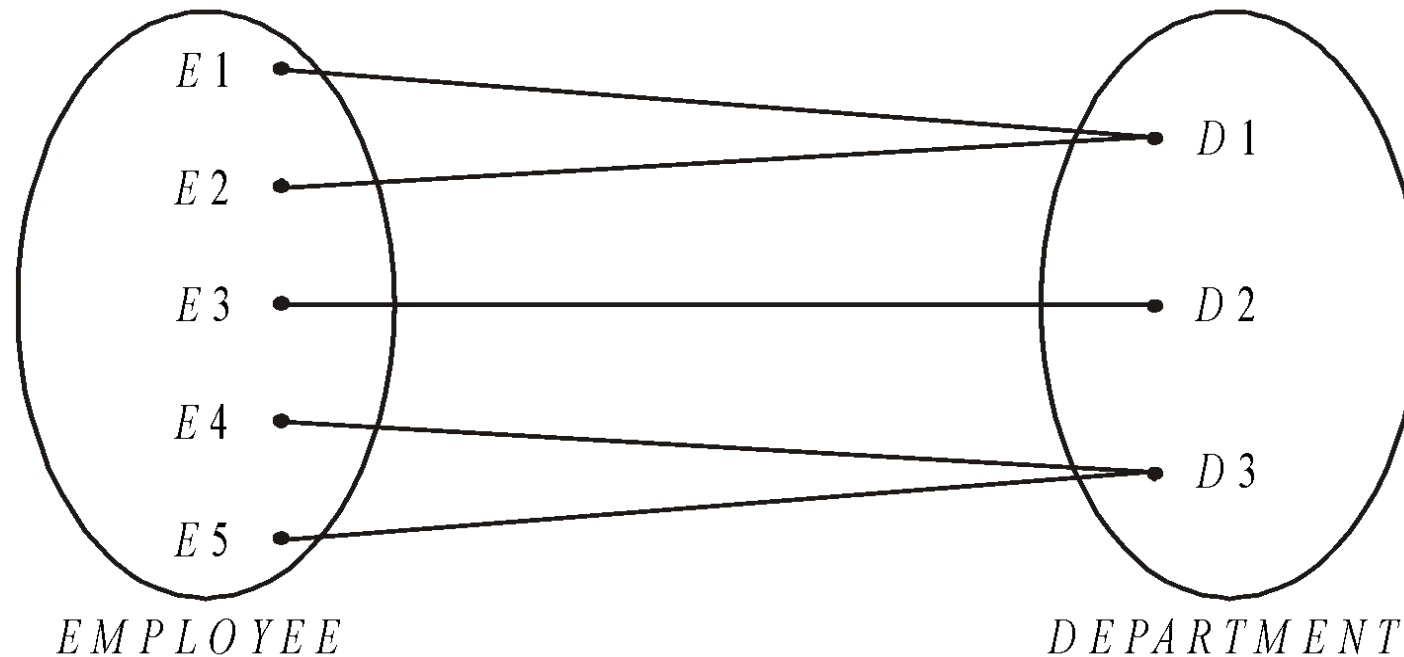
One-to-Many (1:N)

When for one instance of entity A, there are zero, one, or many instances of entity B, but for one instance of entity B, there is only one instance of entity A



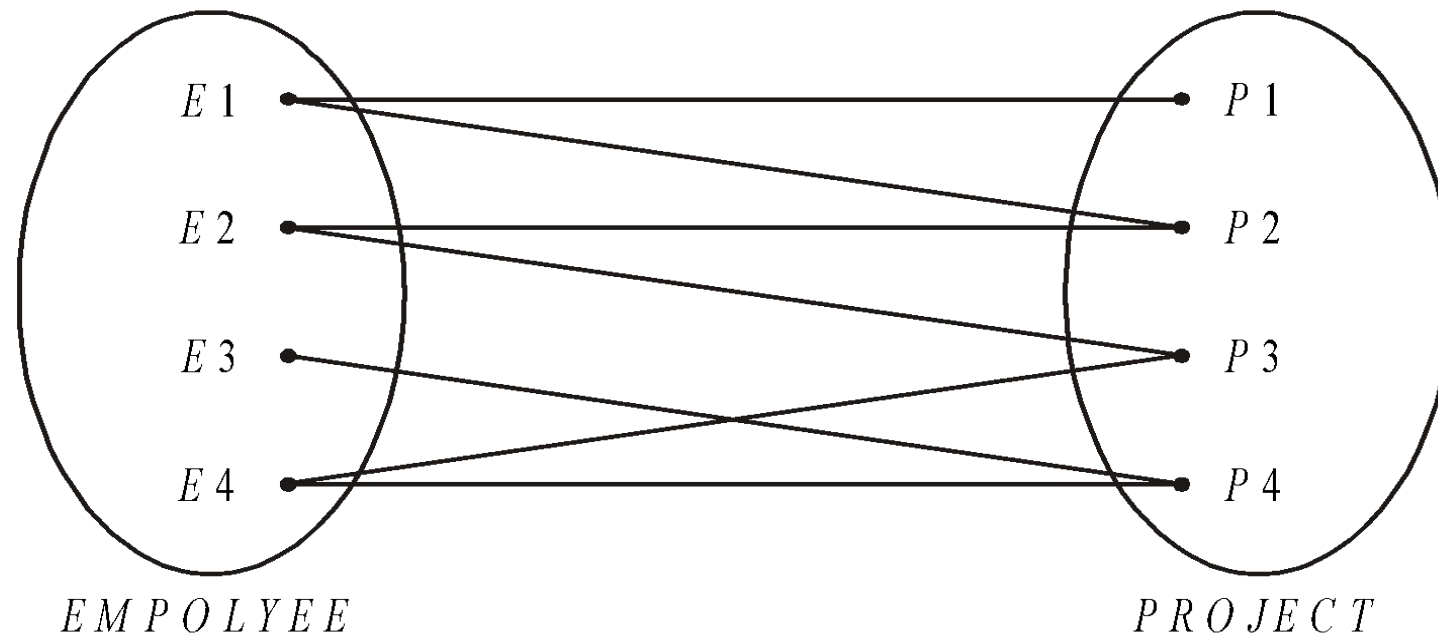
Many-to-One (M:1)

When for zero, one or many instances of entity A, there is only one instance of entity B, but for one instance of entity B, there is only one instance of entity A



Many-to-Many (M:N)

When for one instance of entity A, there are zero, one, or many instances of entity B and for one instance of entity B there are zero, one, or many instances of entity A



Cardinality

The number of instances of one entity that can or must be associated with each instance of another entity.

Minimum Cardinality

If zero, then optional

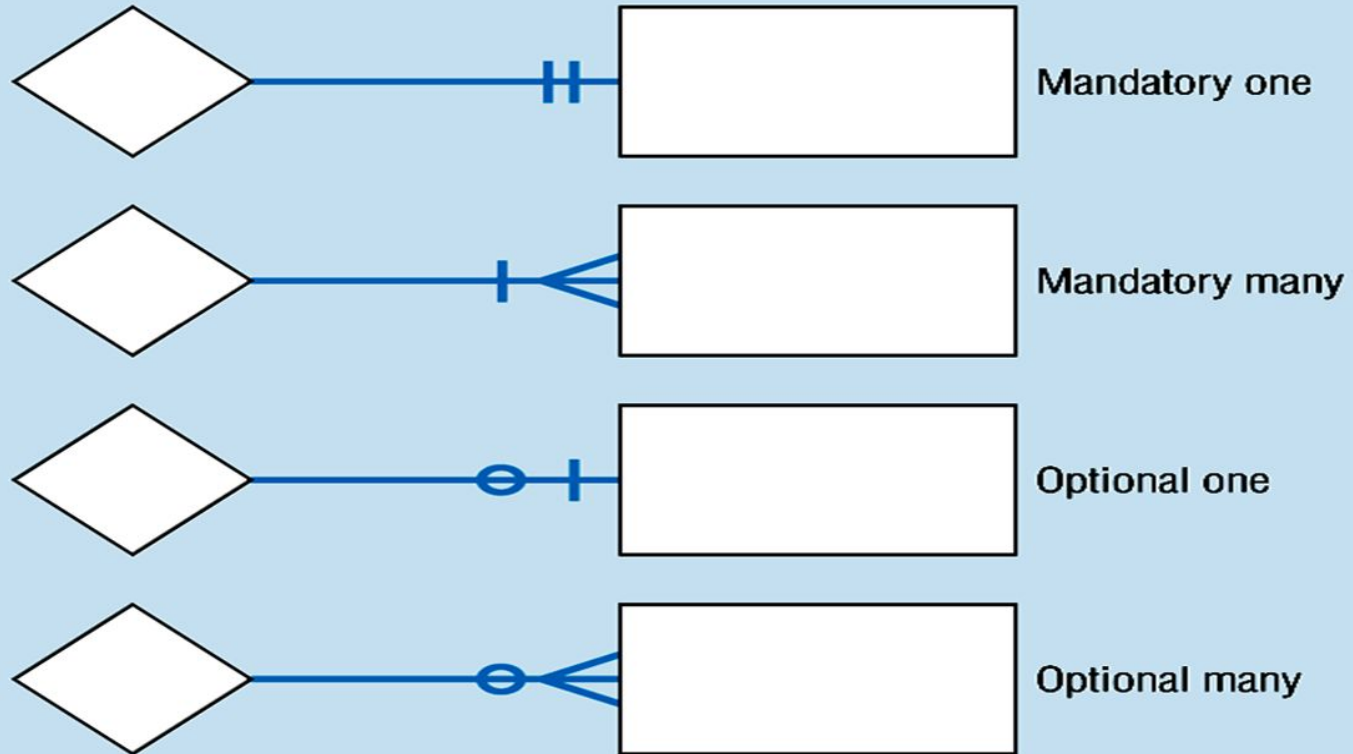
If one or more, then mandatory

Maximum Cardinality

The maximum number

Cardinality

Relationship cardinality



Cardinality

- Minimum of 0 and maximum of 1 relationship



- Minimum of 1 and maximum of 1 relationship



Cardinality

- Minimum of 0 and maximum of >1 relationship



- Minimum of 1 and maximum of >1 relationship



Cardinality

One card holder can have:

0 or 1



Credit card

One card holder can have:

Only 1



Credit card

One card holder can have:

Atleast 0



Credit card

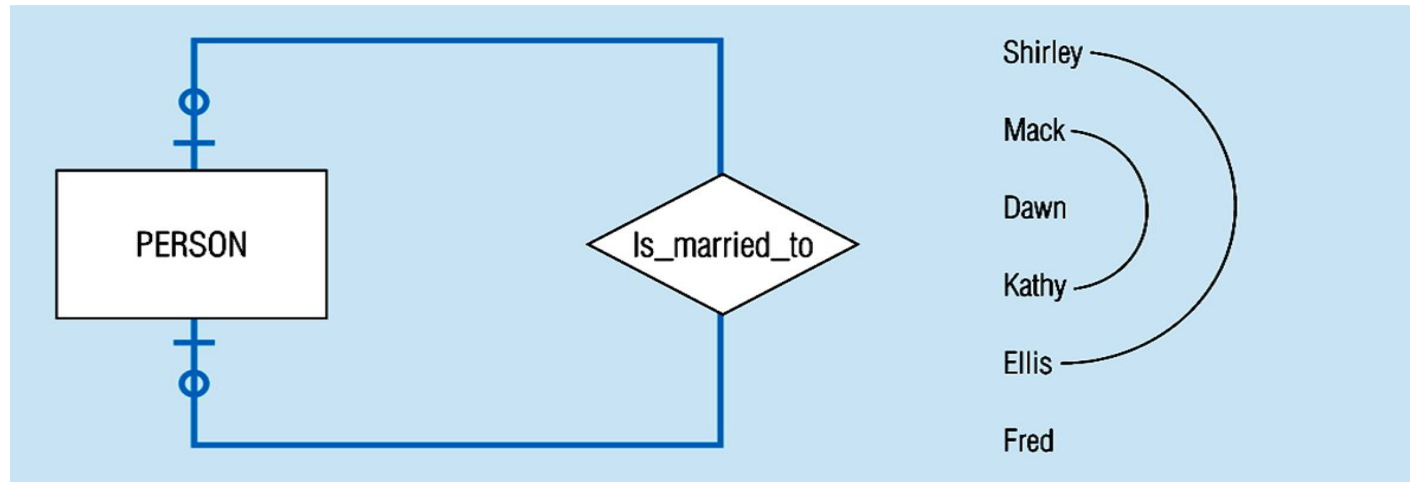
One card holder can have:

Atleast 1



Credit card

Cardinality : Examples

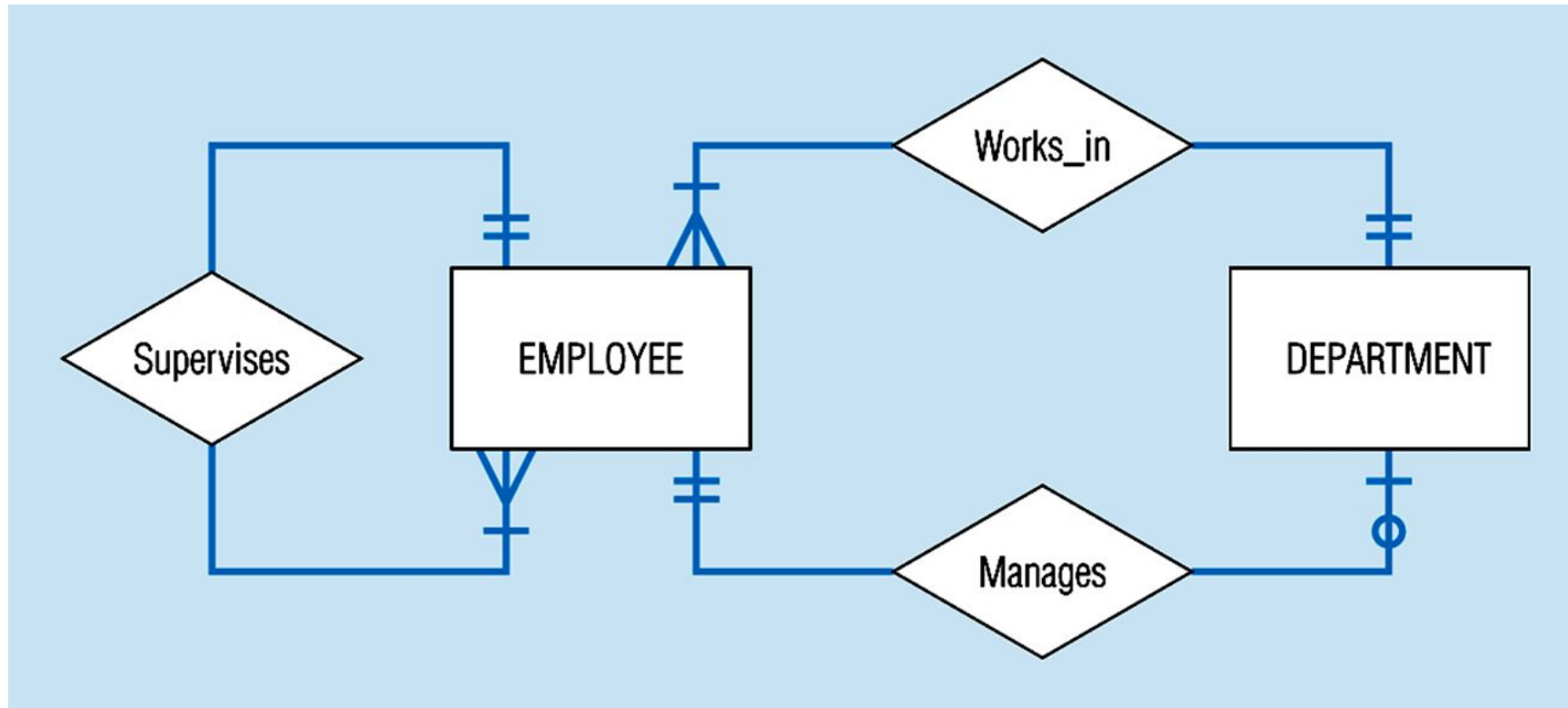


Unary: one to one



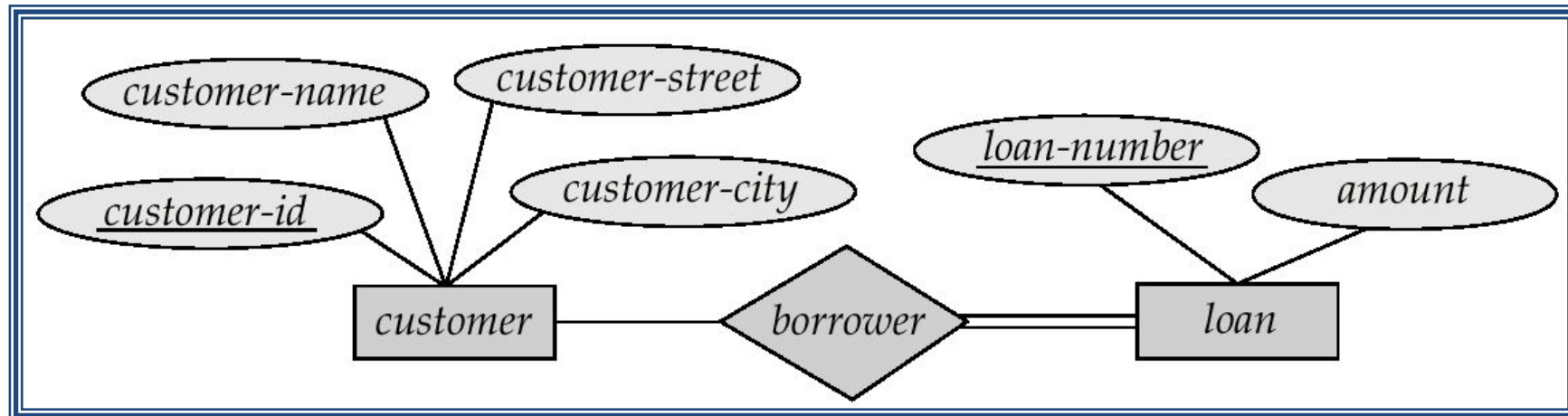
Binary: one to one

Cardinality : Examples

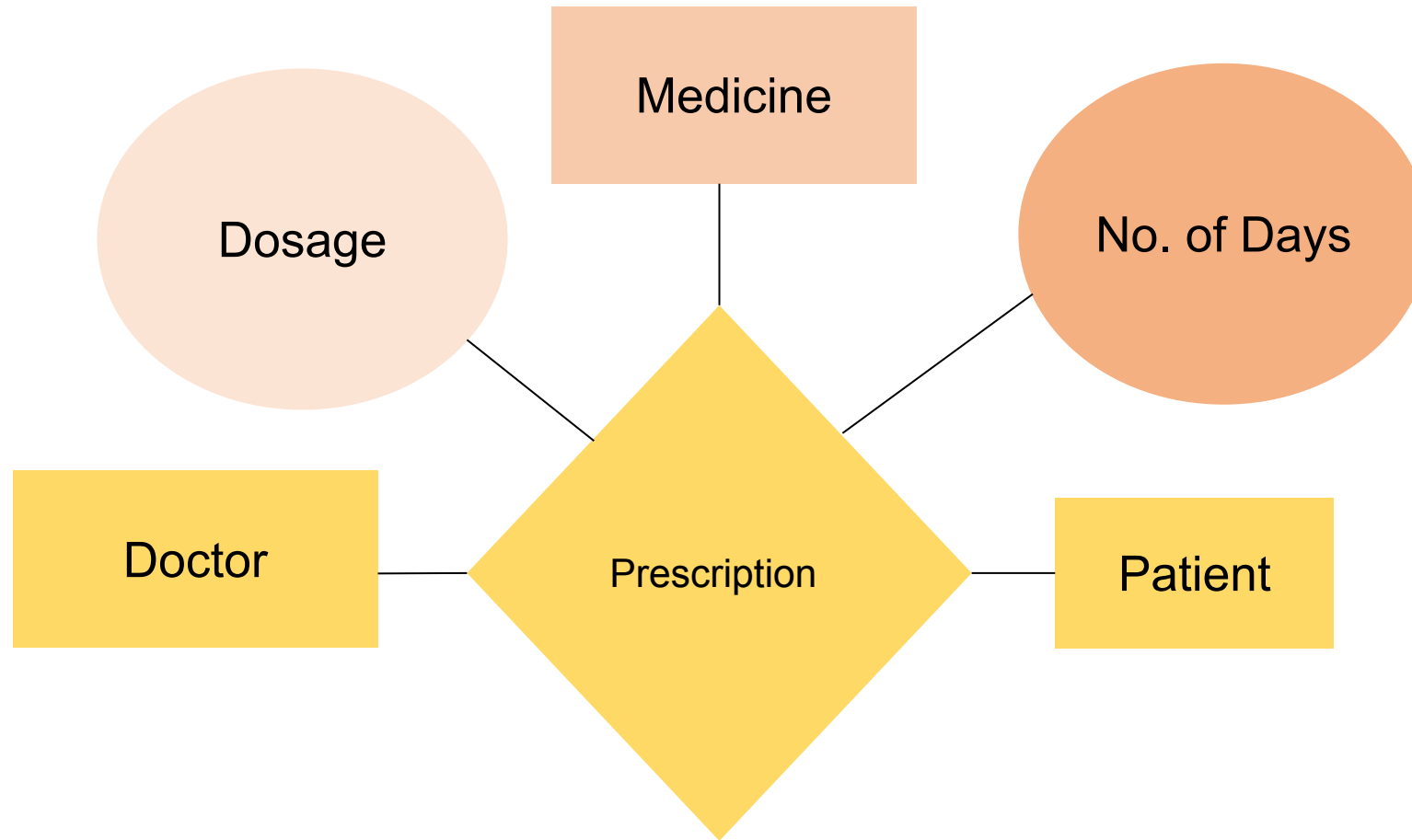


Participation of an Entity Set in a Relationship Set

- Total participation (indicated by double line): every entity in the entity set participates in at least one relationship in the relationship set
 - Participation of loan in borrower is total (every loan must have a customer associated to it via borrower)
- Partial participation(indicated by single line): some entities may not participate in any relationship in the relationship set
 - Participation of customer in borrower is partial



Attribute of a Relationship



Dependent and Independent Entities

- Entity B is existence-dependent on entity A when
 - Some instance of A must exist before B can exist
 - If A ceases to exist, B must also cease to exist
- An independent entity is not existence-dependent on any other entity
- Independent Entity – Strong Entity
- Dependent Entity – Weak Entity

Strong vs. Weak Entities

Strong entities

- Exist independently of other types of entities
- Has its own unique identifier
- Represented with single-line rectangle

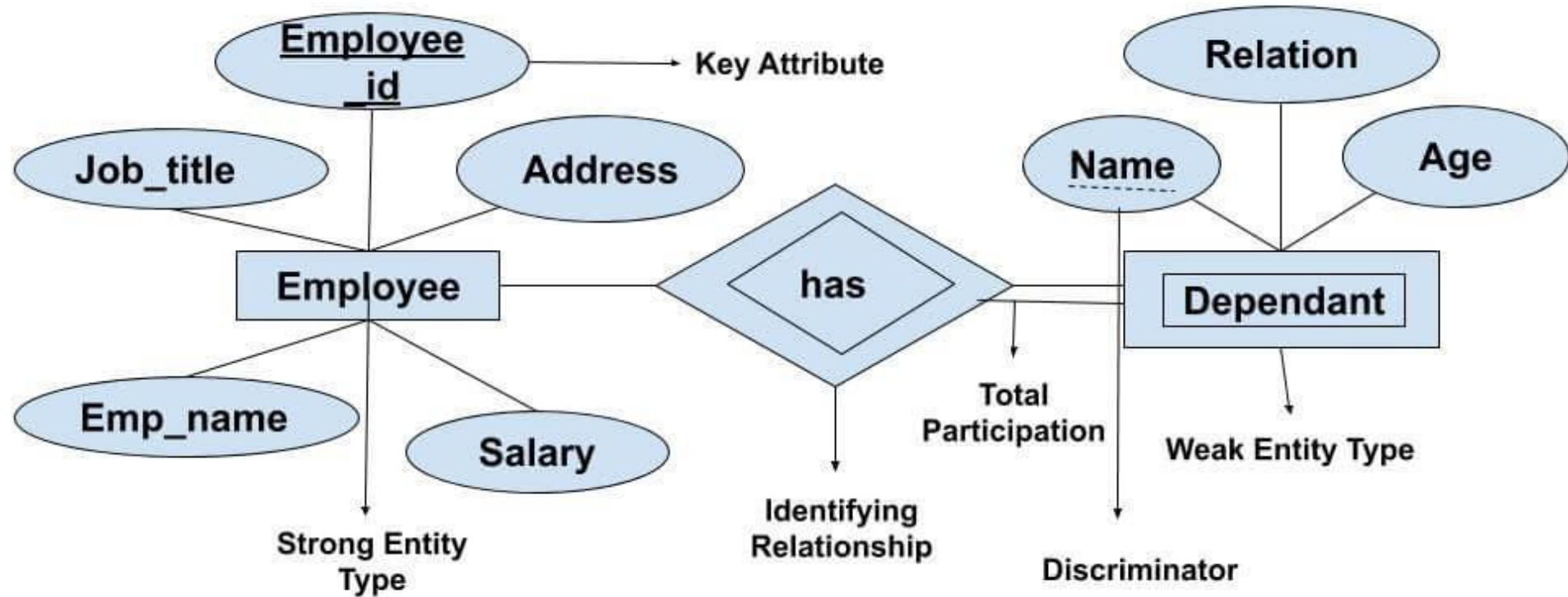
Weak entity

- Dependent on a strong entity
- Cannot exist on its own
- Unique identifier is called Discriminator
- Represented with double-line rectangle

Identifying relationship

- Links strong entities to weak entities
- Represented with double line diamond

Strong vs. Weak Entities



Designing ERD

Steps in ER Modeling

Step 1: Identify the Entities

Step 2: Find the relationships

Step 3: Identify the key attributes

Step 4: Identify other relevant attributes

Step 5: Draw complete E-R diagram with all attributes including Primary Key

Designing ERD

Assumptions :

There are multiple banks and each bank has many branches.

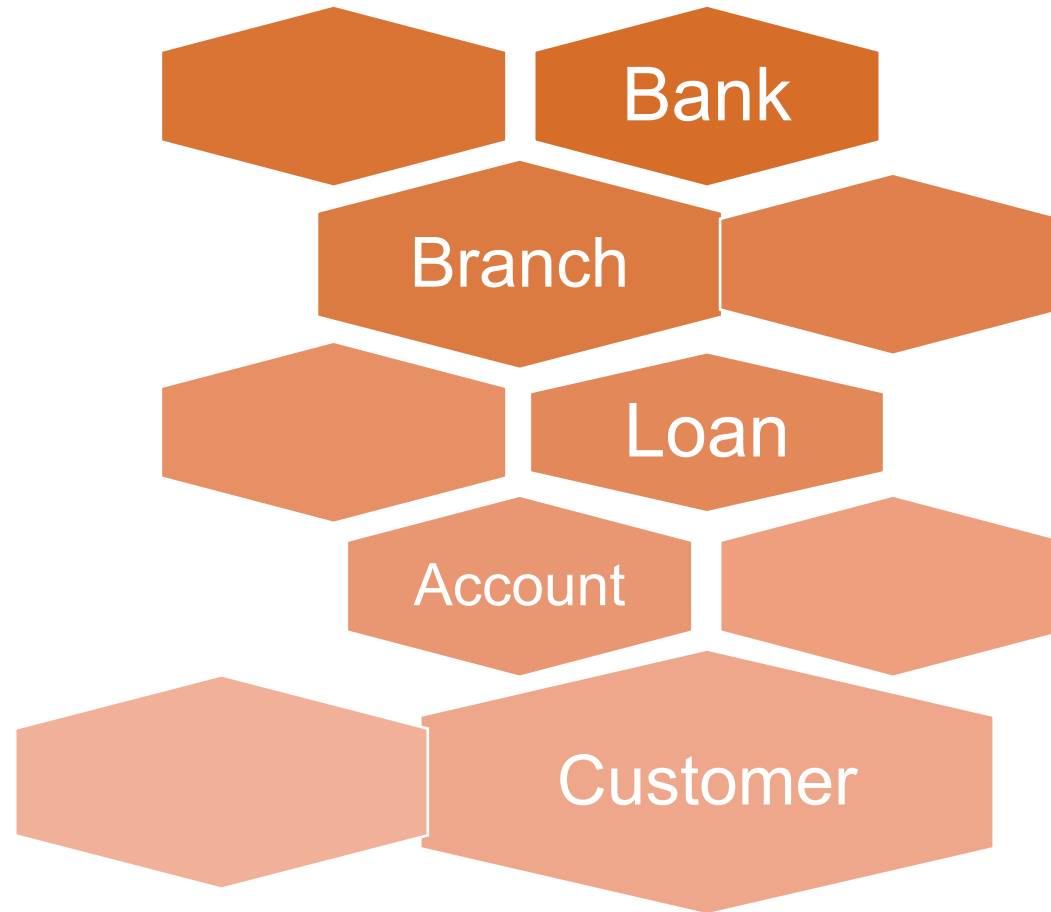
Each branch has multiple customers

Customers have various types of accounts in any branch

Some Customers also had taken different types of loans from these bank branches

One customer can have multiple accounts and Loans

Step 1: Identify the Entities



Step 2: Find the Relationships

- One Bank has many branches and each branch belongs to only one bank, hence the cardinality between Bank and Branch is One to Many.
- One Branch offers many loans and each loan is associated with one branch, hence the cardinality between Branch and Loan is One to Many.
- One Branch maintains multiple accounts and each account is associated to one and only one Branch, hence the cardinality between Branch and Account is One to Many
- One Loan can be availed by multiple customers, and each Customer can avail multiple loans, hence the cardinality between Loan and Customer is Many to Many.
- One Customer can hold multiple accounts, and each Account can be held by multiple Customers, hence the cardinality between Customer and Account is Many to Many

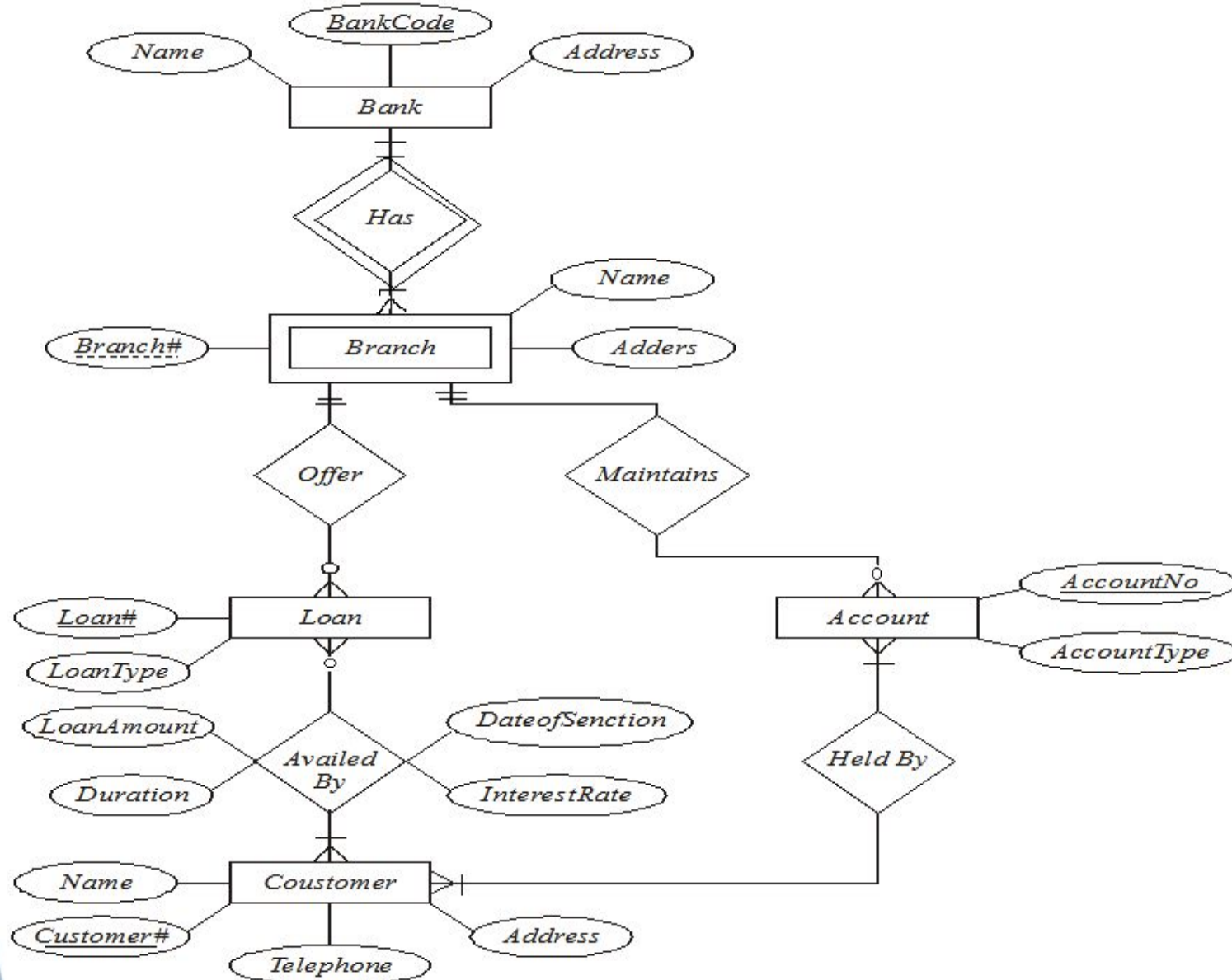
Step 3: Identify the Key Attributes

- BankCode (Bank Code) is the key attribute for the Entity “Bank”, as it identifies the bank uniquely.
- Branch# (Branch Number) is the key attribute for “Branch” Entity.
- Customer# (Customer Number) is the key attribute for “Customer” Entity.
- Loan# (Loan Number) is the key attribute for “Loan” Entity.
- Account# (Account Number) is the key attribute for “Account” Entity.

Step 4: Identify Other Relevant Attributes

- For the “Bank” Entity, the relevant attributes other than “BankCode” would be “Name”
- For the “Branch” Entity, the relevant attributes other than “Branch#” would be “Name”
- For the “Loan” Entity, the relevant attribute other than “Loan#” would be “Loan Type”.
- For the “Customer” Entity, the relevant attributes other than “Customer#” would be “Name”, “Telephone#” and “Address”.
- For the “Account” Entity, the relevant attribute other than “Account No” would be “Account Type”.

Bank ERD



Thank you!

