ER Model

The ER Model…

- ER model stands for Entity Relationship model.
- This model is very useful for conceptual database design because of its close relationship with the real world that it models.
- This model is conveniently represented by a diagram called the ER diagram.

…The ER Model…

- An ER model tries to represent some portion of the real world in the computer. This portion is known as the "miniworld".
- The basic elements of an ER model are entities, attributes and relationships.

Entities

- Entities represent a “thing” in the miniworld with an independent existence.
- These may be objects with physical existence (e.g. a person, a car, a house, an employee) or objects with conceptual existence (e.g. a company, a job, a college course)

Notation for an Entity

- An entity in an ER diagram is represented by a rectangle.

Attributes…

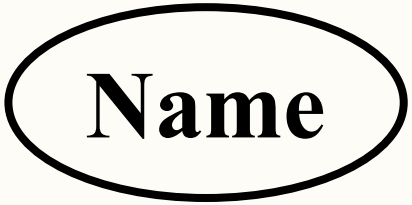
- An entity may have several attributes. These attributes describe the characteristics of the entity. E.g. An entity Person may have attributes like Name, Sex, Address, Phone, Birthdate etc.

…Attributes…

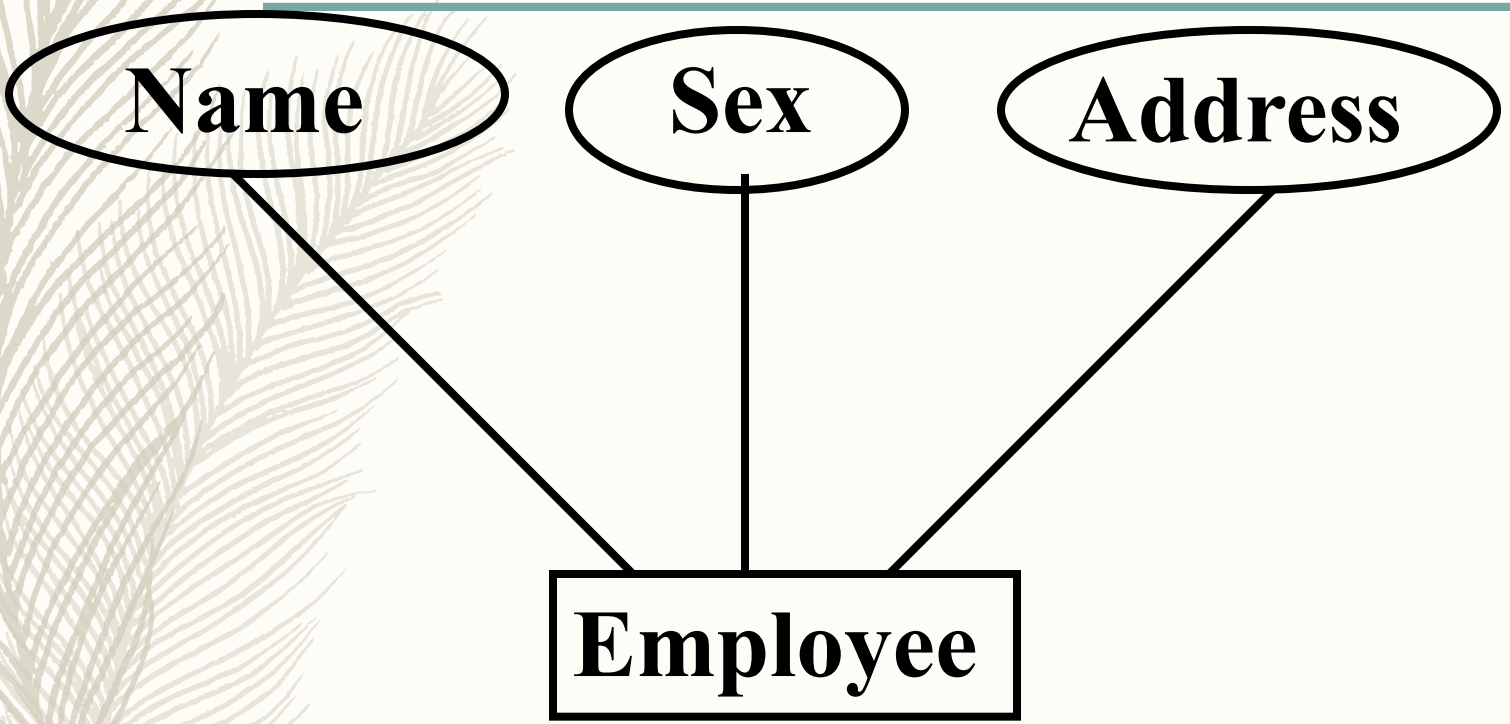
- An entity Car may have attributes like Manufacturer, Model, color, YearOfManufacture, RegistrationNumber etc.
- An entity Computer may have attributes like Make, Processor, Speed, RAM, Hard Disk etc.

Notation for Attributes

- An attribute is represented by an oval and is connected to its entity by a straight line.



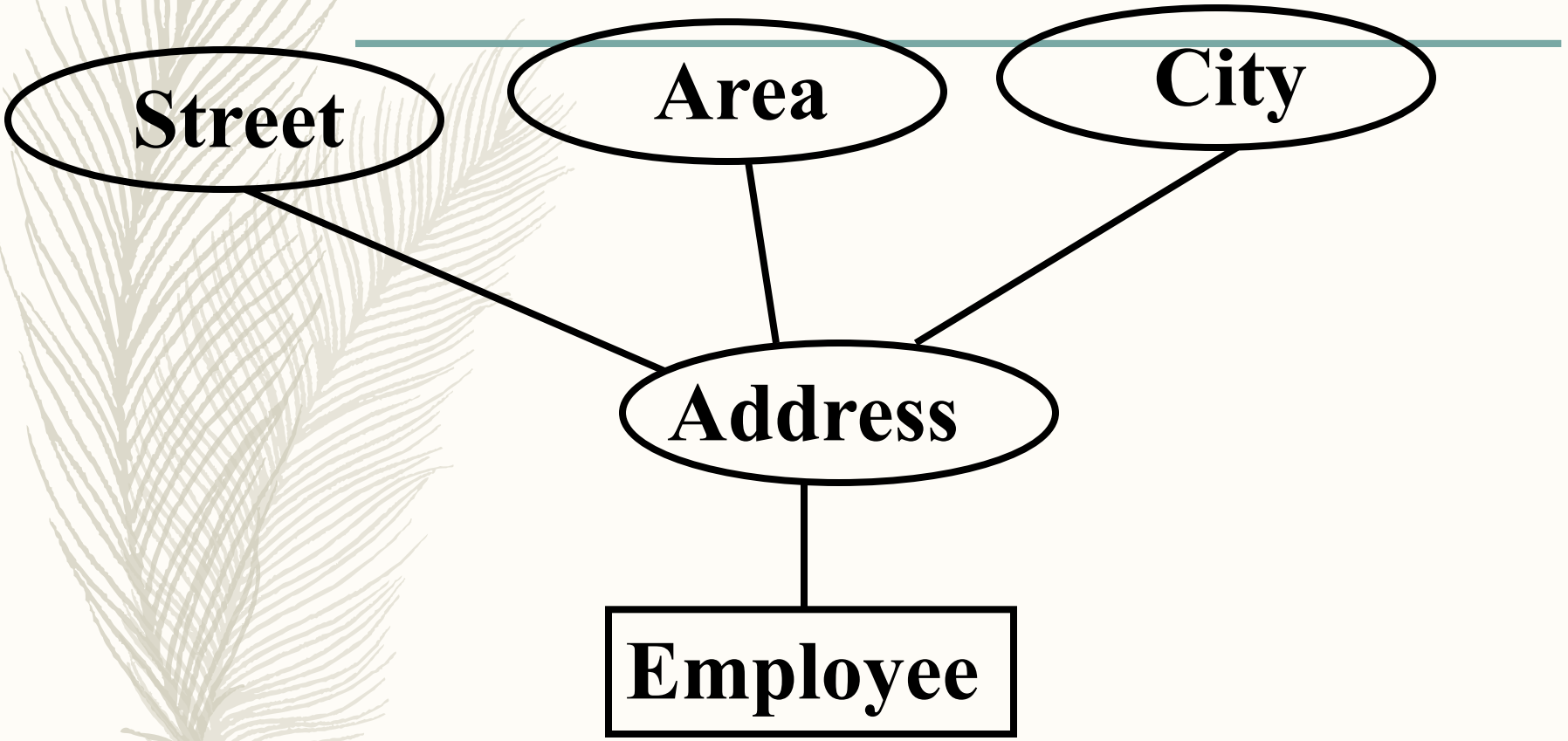
An Entity With Attributes



Composite Versus Simple Attributes

- A composite attribute is composed of several subattributes.
- E.g. the attribute Address may be composed of (StreetAddress, Area, City, State, PINCode).
- Non-composite attributes are called simple or atomic attributes.

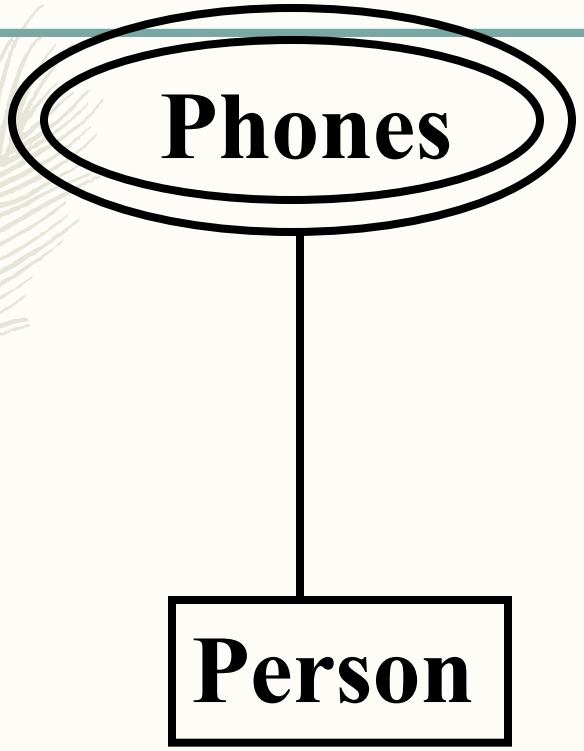
A Composite Attribute



Single-Valued V/S Multivalued Attributes

- A single valued attribute can have only one value (e.g. Name of a Person), while a multivalued attribute can have several values (e.g. Phones Attribute of a Person).

Notation for a Multivalued Attribute



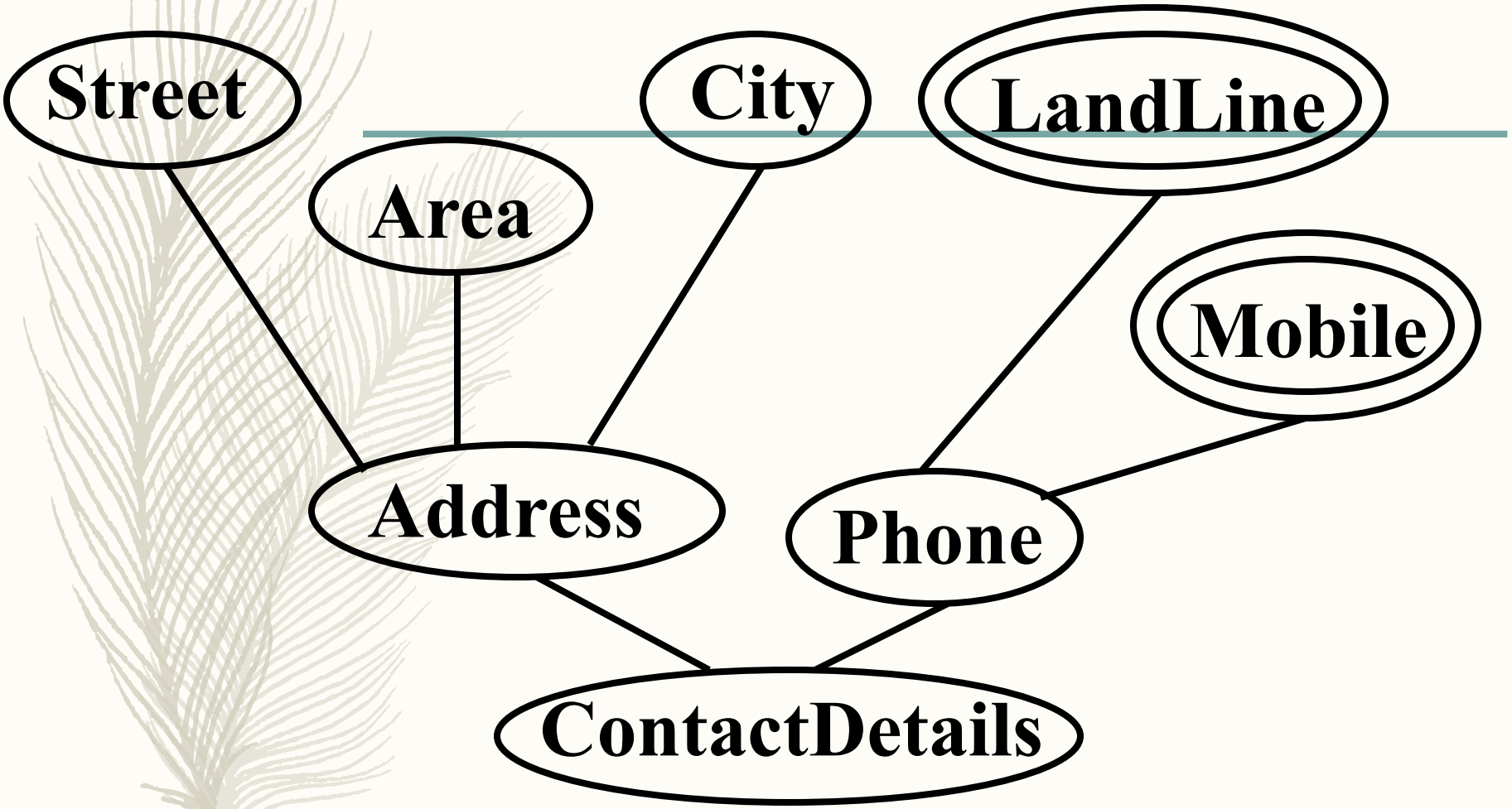
Complex Attributes

- Composite and multivalued attributes can be combined and nested one inside another up to any arbitrary level, resulting in complex attributes.

Example of a Complex Attribute

- The attribute ContactDetails may have two subattributes : Address (a composite attribute as described earlier) and Phone.
- Phone may have two subattributes: LandLine and Mobile, each a multivalued attribute.

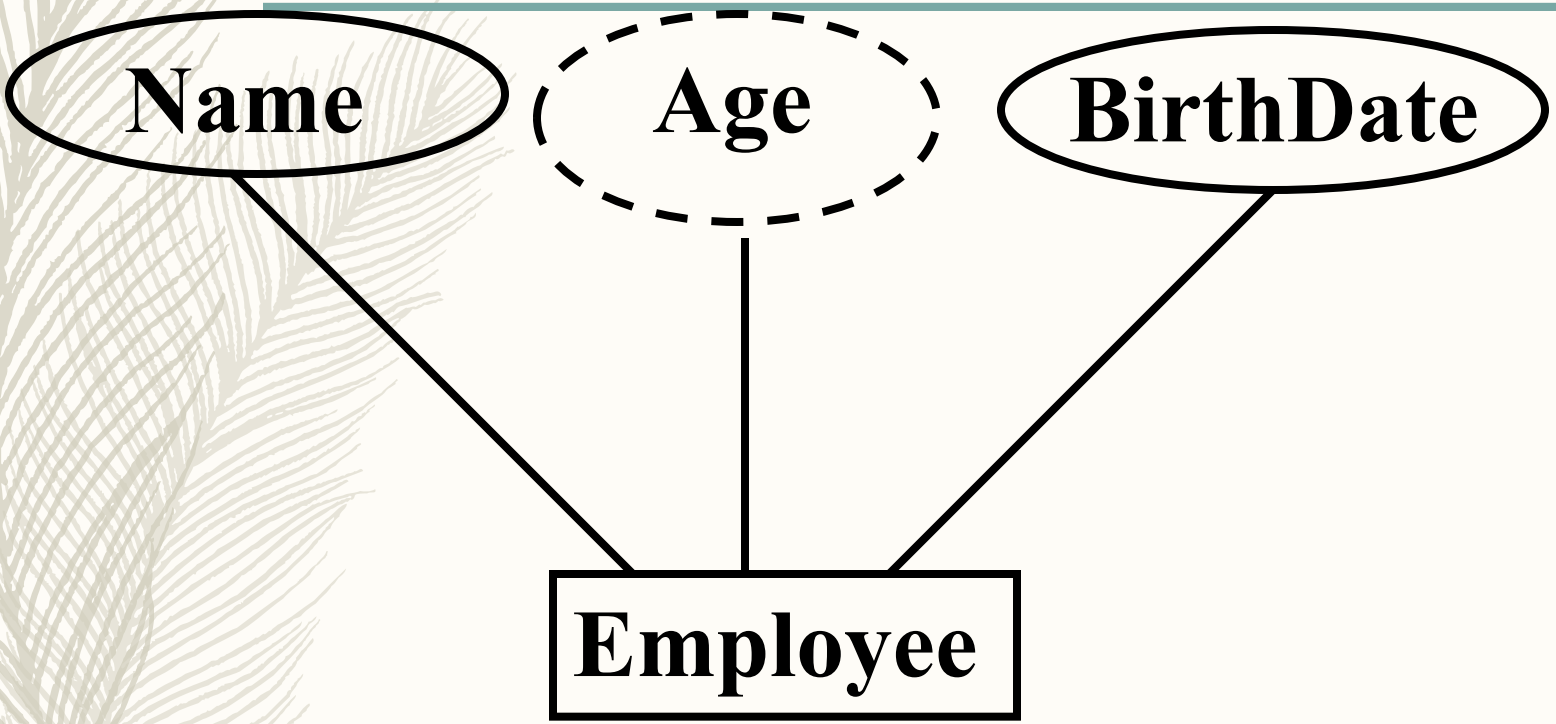
Example of a Complex Attribute



Stored versus Derived Attributes

- A stored attribute is one that cannot be derived from other attributes (e.g. BirthDate attribute of a Person)
- Derived attribute is one whose value can be obtained from other attributes (e.g. Age attribute of a Person)

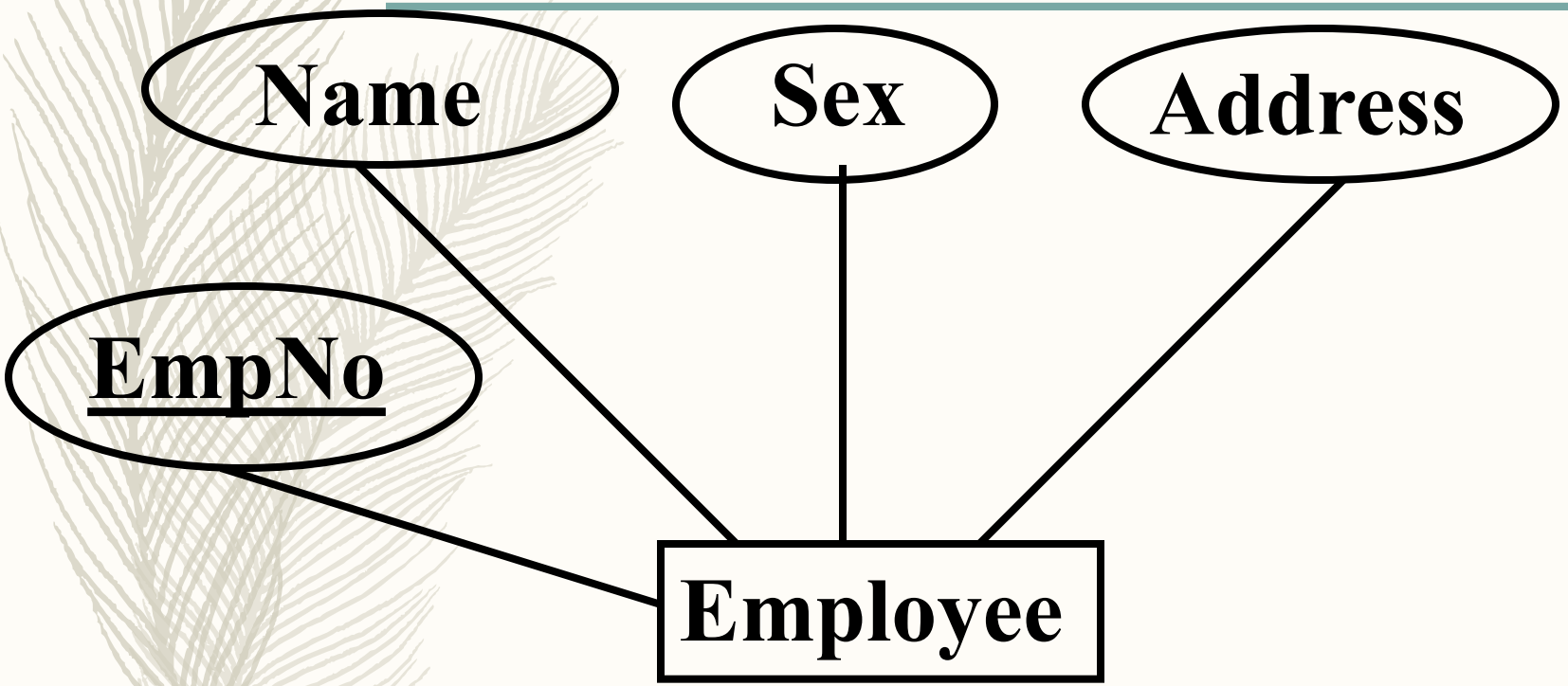
Notation for a Derived Attribute



Entities and Keys

- Usually, most entities have some combination of their attributes as the *key* for that entity.
- Every entity in an entity set has a distinct value for its key attribute(s).
- An entity having a key is known as a *strong entity*.

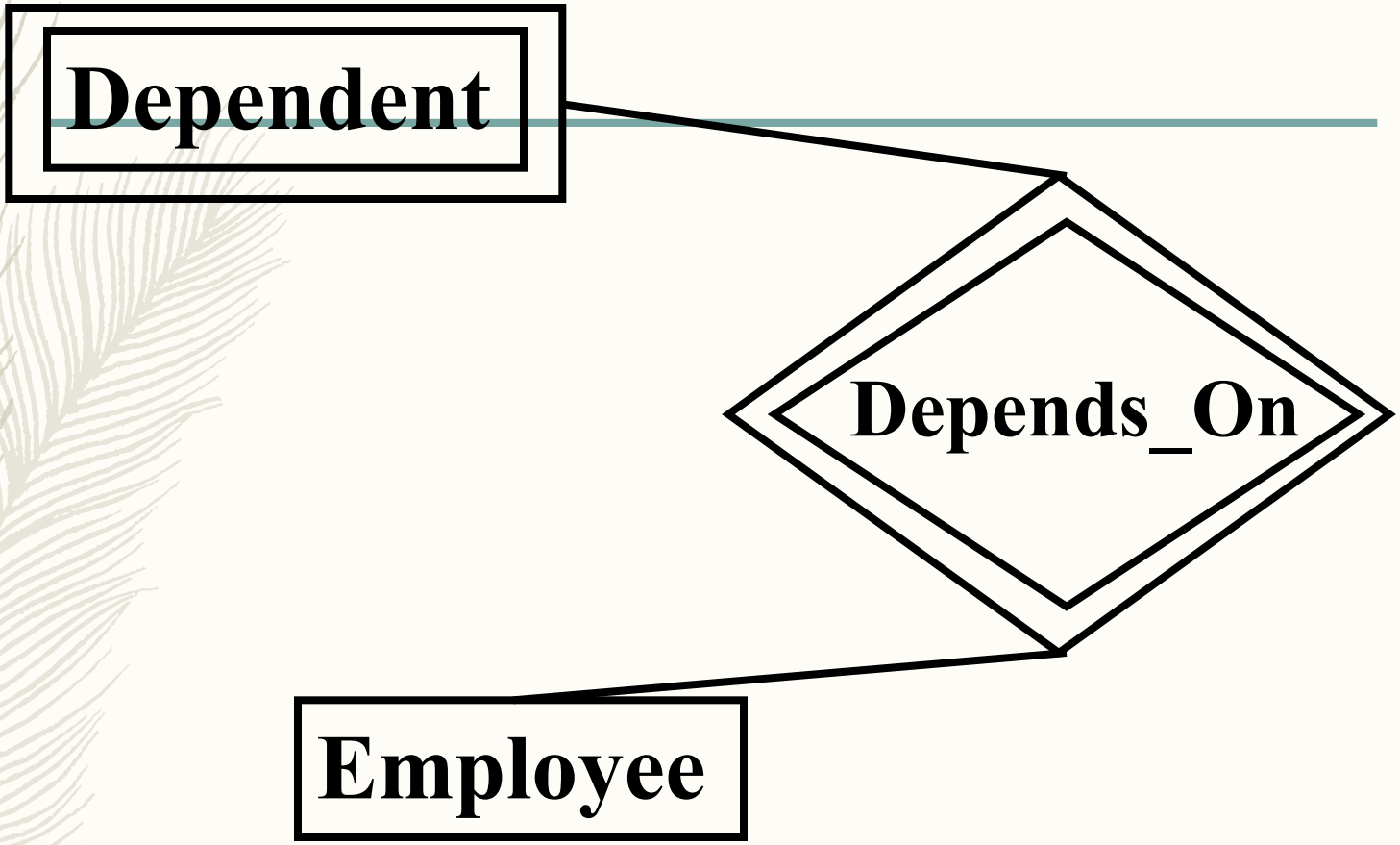
Notation for a Key Attribute



Weak Entities

- An entity that doesn't have a key is known as a weak/subordinate/child entity.
- Such an entity is always related with another entity called the strong/dominant/parent entity so that some attribute(s) of the weak entity, together with the key of the parent entity always have a distinct value.

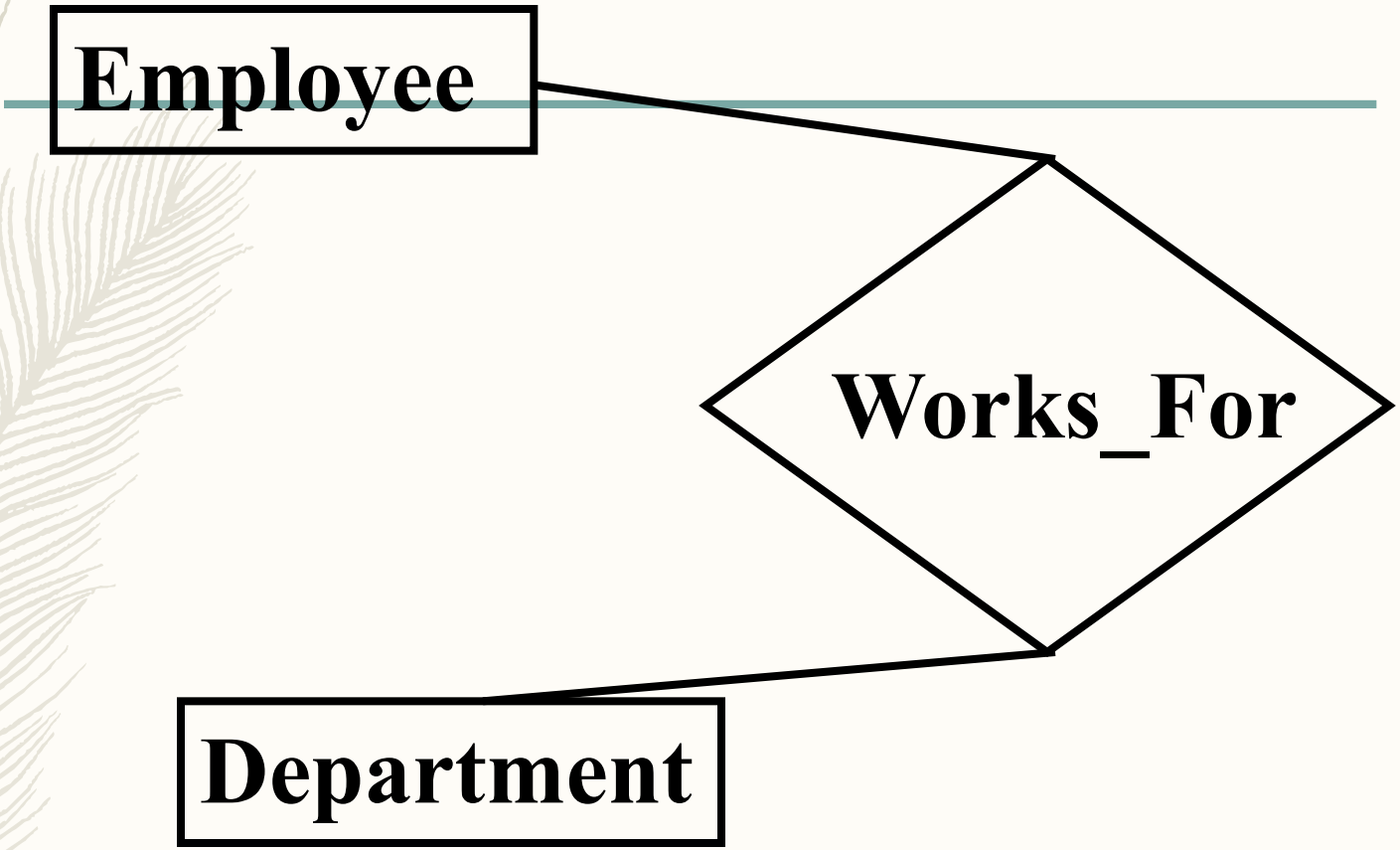
Notation for a Weak Entity



Relationships

- Two entities may be connected by a relationship.
- For example, an Employee is connected to a Department by the relationship Works_For.
- A Book is connected to a Member by the relationship Issued_To.

Notation for a Relationship



Relationship Types

- There are three types of relationships:
  - one-one (1:1),
  - one-many (1:M) (or many-one (M:1))
  - many-many (M:N).
- These are also known as cardinality ratios for relationships.

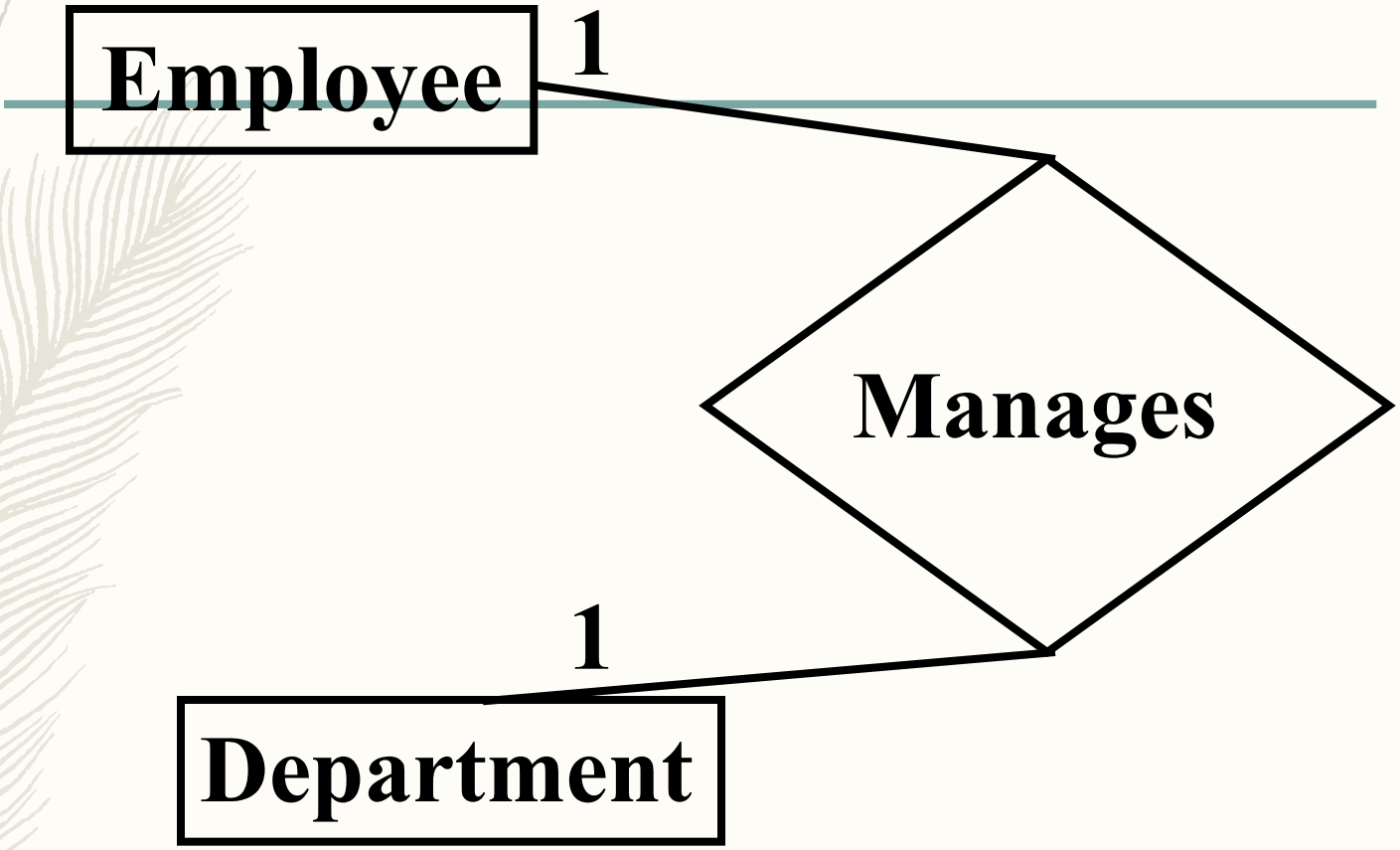
One-One Relationship

- In this relationship exactly one entity on one side of the relationship is associated with exactly one entity on the other side of the relationship.

Example of One-One Relationship

- The Employee and Department entities are related by the relationship Manages (i.e. an Employee manages a Department).
- Only one Employee manages any given Department and any given Employee can manage only one Department.

Notation for One-One Relationship



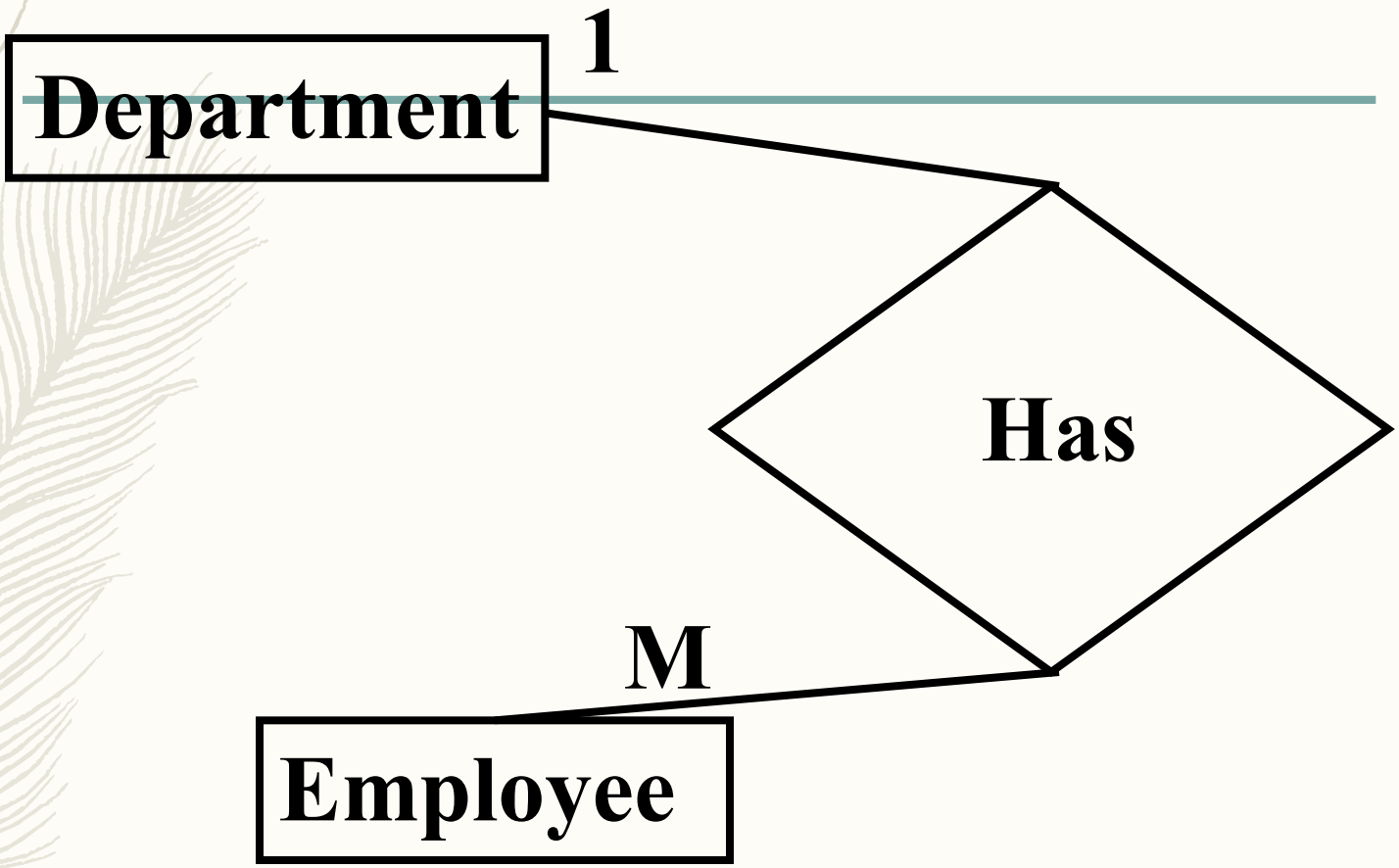
One-Many Relationship

- In this relationship exactly one entity on one side of the relationship is associated with zero or more entities on the other side of the relationship.

Example of One-Many Relationship

- One Department may have many Employees working for it.

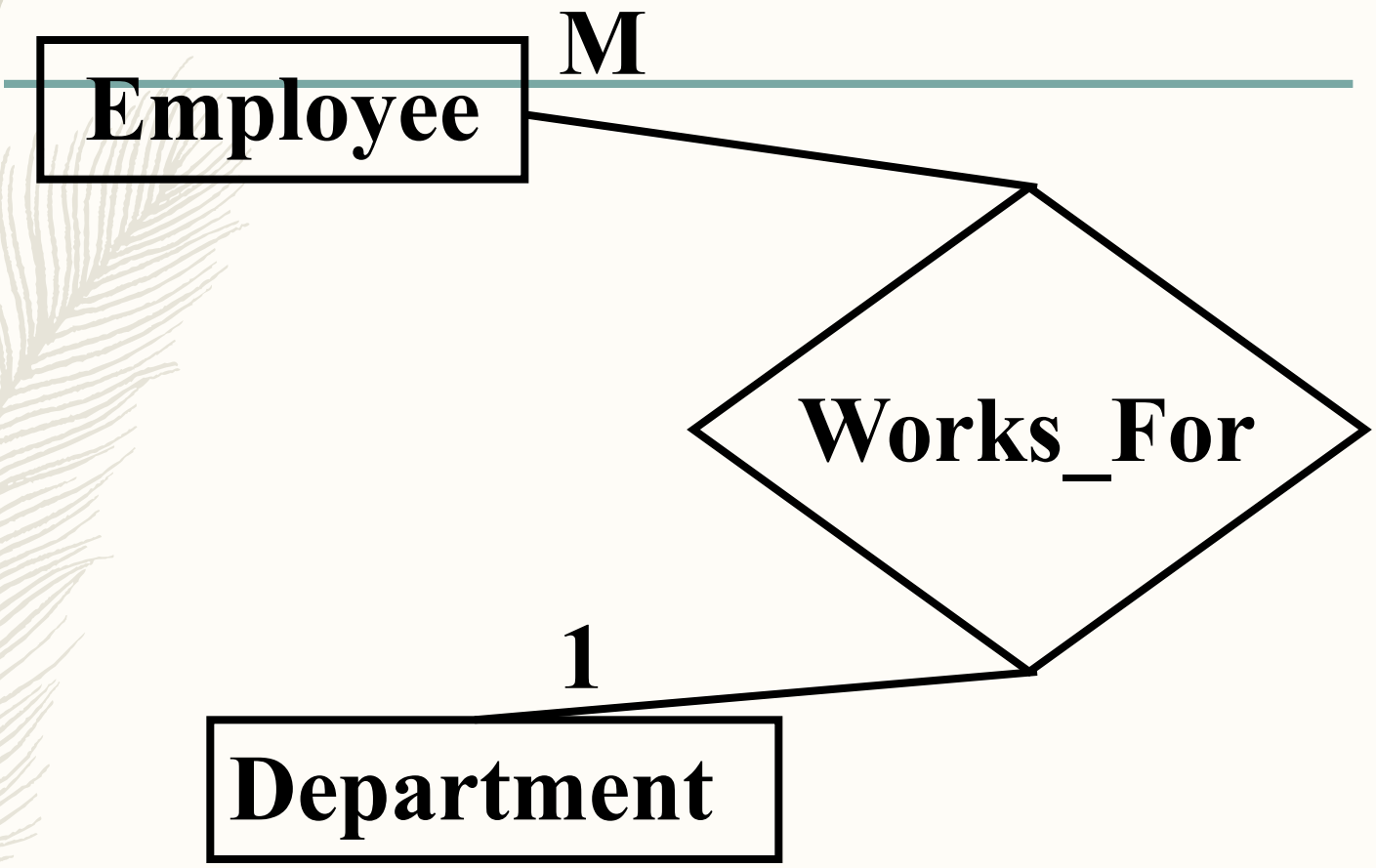
Notation for One-Many Relationship



Many-One Relationship

- It is nothing but a One-Many relationship with its sides reversed.
- E.g. Many Employees may be working for one Department.

Notation for Many-One Relationship



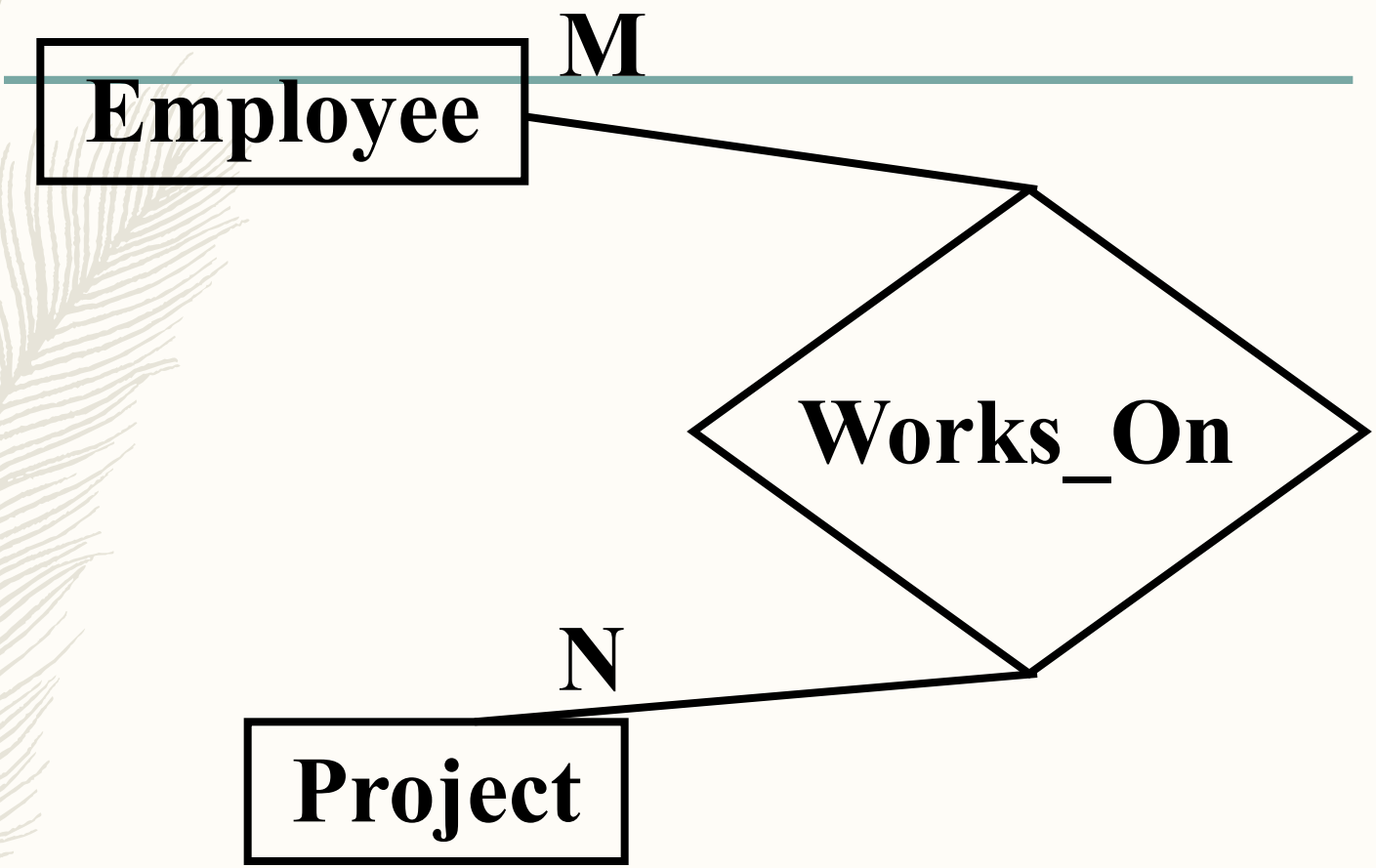
Many-Many Relationship

- In this kind of relationship, zero or more entities on one side may be related with zero or more entities on the other side.

Example of Many-Many Relationship

- One Employee may work on several Projects at a time and one Project may have several Employees working on it.

Notation for Many-Many Relationship



Relationship Attributes

- A relationship can also have attributes.
- E.g. the Works_On relationship in the preceding example may have an attribute Hours indicating how many hours a particular employee works on a Project.

Notation for Relationship Attributes