ER Model in DBMS

Entity + Relationship + Model

Why ER Model???

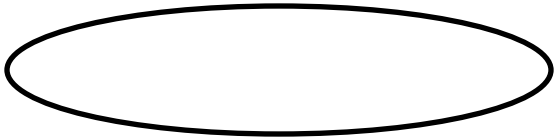
ER Model

- The ER Model was developed by “Peter Chen” and published in a 1976 paper.
- This model is a high level conceptual data model.
- ER model is a **Visual Representation** of data that describe how data is related to each other.
- It is based on perception of real world that consist of a collection of object(called entities) and of relationship among these objects.

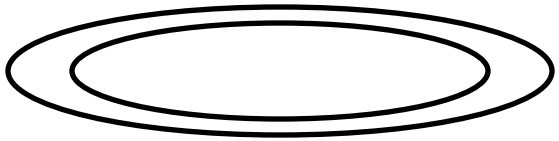
Basic Terminology of ER Model

Entity or Strong Entity

Weak Entity

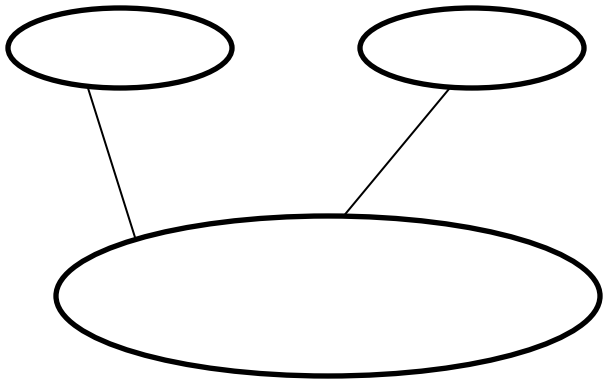


Attribute

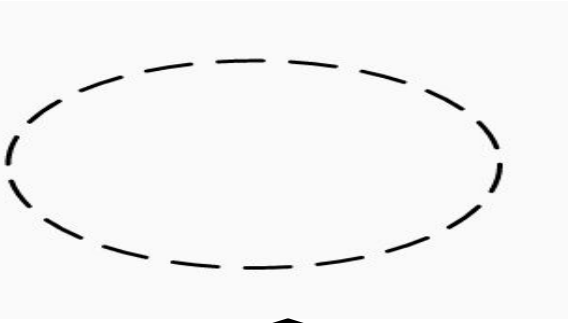


Multivalued Attribute

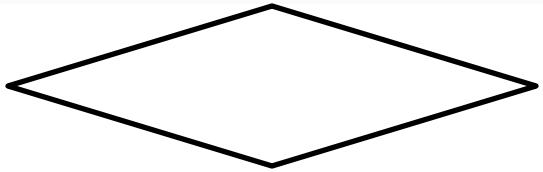
Basic Terminology of ER Model



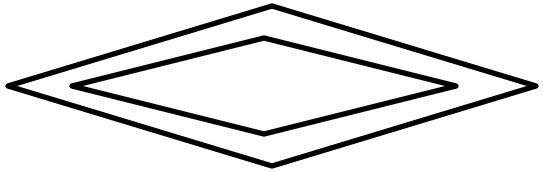
Composite Attribute



Derived Attribute



Relationship



Weak Relationship

Basic Terminology of ER Model

- **Entity**:
  - A entity is a “thing” or “object” in the real world.
  - An entity contains **attributes**, which describe to that entity. So anything about which we store information is called an entity.
  - For example:
    - A Student, An employee, or bank a/c etc.

Basic Terminology of ER Model

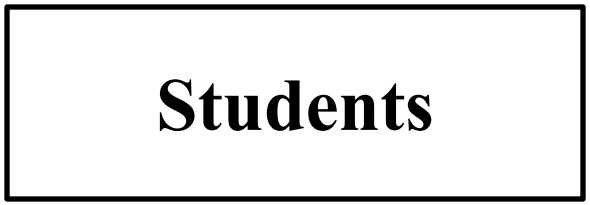
Students

Roll No.	Name	DOB	Course	Address
1	Prashant	21.1.1993	MCA	Delhi
2	Vipul	12.5.1994	BCA	Punjab
3	Nilam	10.9.1994	MCA	Delhi
4	Nidhi	20.1.1993	MCA	Punjab

Types of Entity

- **Strong Entity**
- **Weak Entity**

- **Strong Entity**: An Entity having an attributes, which can be used as candidate key or primary key is called as strong entity.



Types of Entity

- **Weak Entity:** A weak entity does not have any primary key means the entity which does not any sufficient attributes to form a primary key.

Basic Terminology of ER Model

- **Attributes**:

  - Attributes are the properties of an entity, which describe characteristics of that entity.
  - Types of attributes:

    I. Simple and Composite A.
    II. Single value and Multivalued Attributes
    III. Stored and Derived Attributes

Types of Attributes

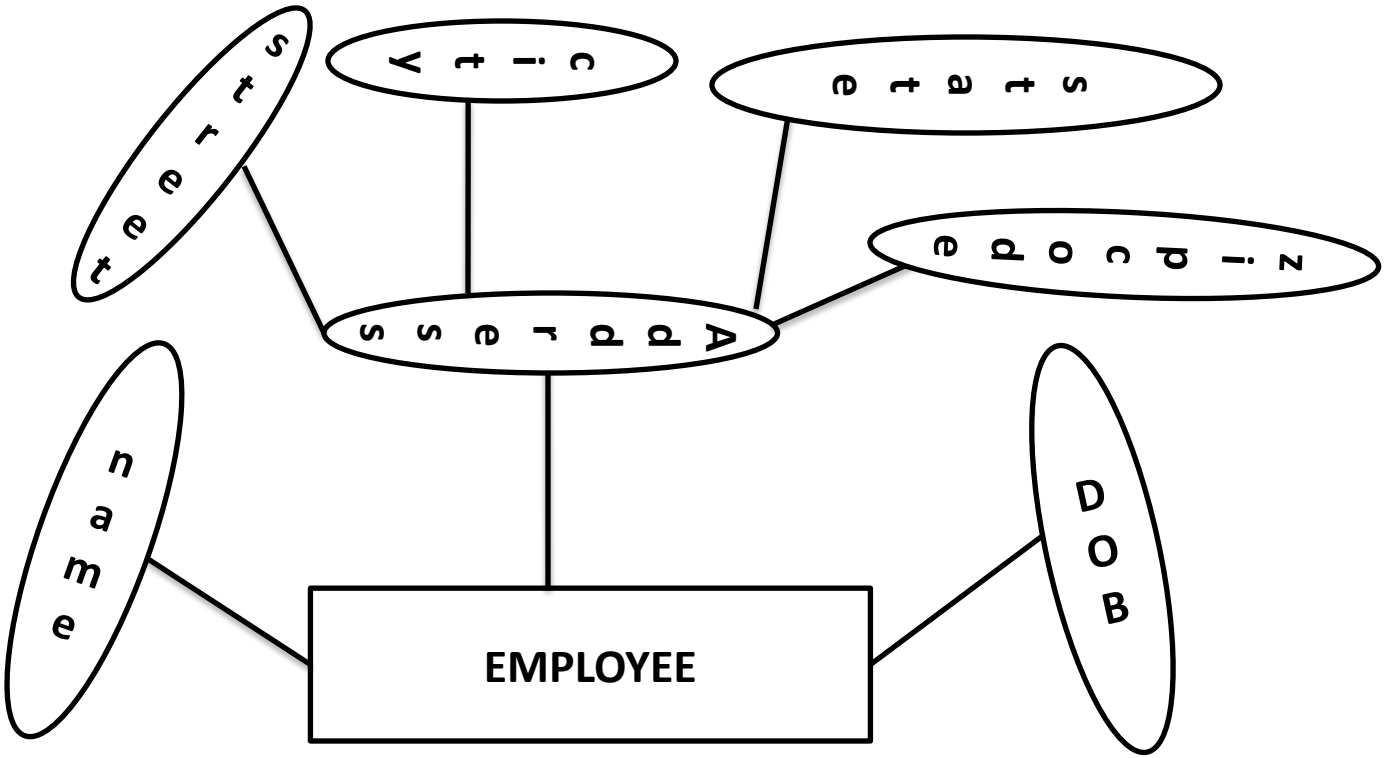
I. Simple and Composite Attributes:

- A **simple attributes** is just an attributes that can not be divide into parts.
- Example: name, age, salary etc.

- A **composite attribute** is an attribute that can be subdivided into other attributes.
- Example: “Address” can be subdivided into street, city, state, zip code etc.

Types of Attributes

I. Simple and Composite Attributes:

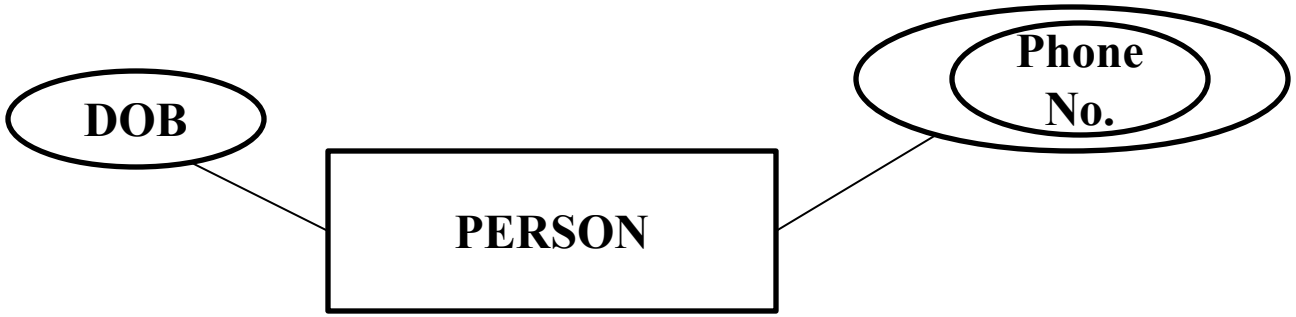


Types of Attributes

II. Single value and Multivalued Attributes:

- A **single value** attribute can have only a single value
- Example:
  - A person have only one “DOB” or “age”
  - A student can have only one “roll-no”

- A Multivalued attributes can have multiple value
- Example: A person may have multiple phone numbers



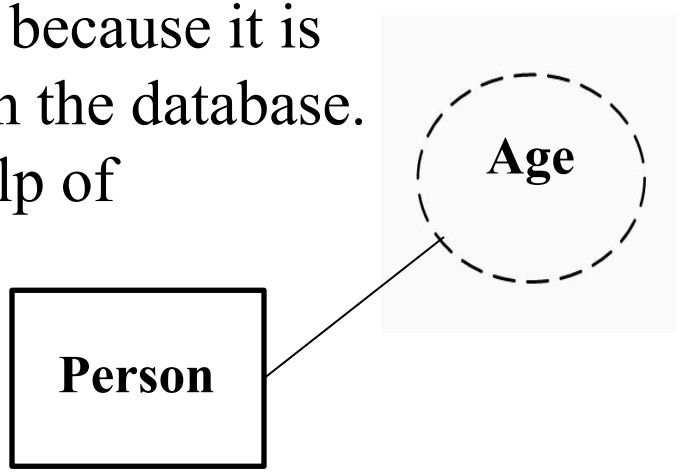
Types of Attributes

III. Stored and Derived Attributes

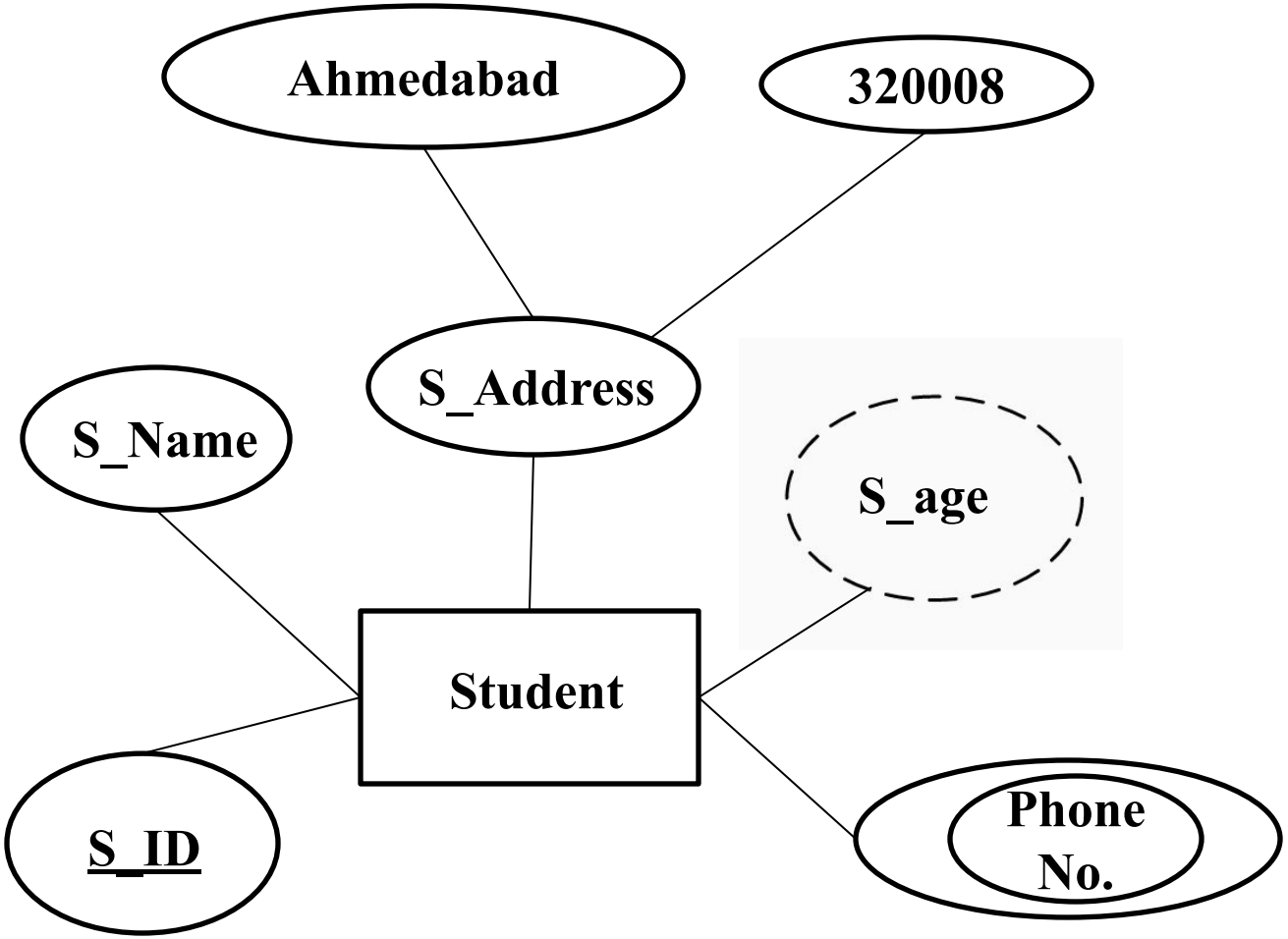
- Stored attribute are those attributes, which are physically stored into the database.

- Derived attributes are those attributes, which are not physically stored into the database.

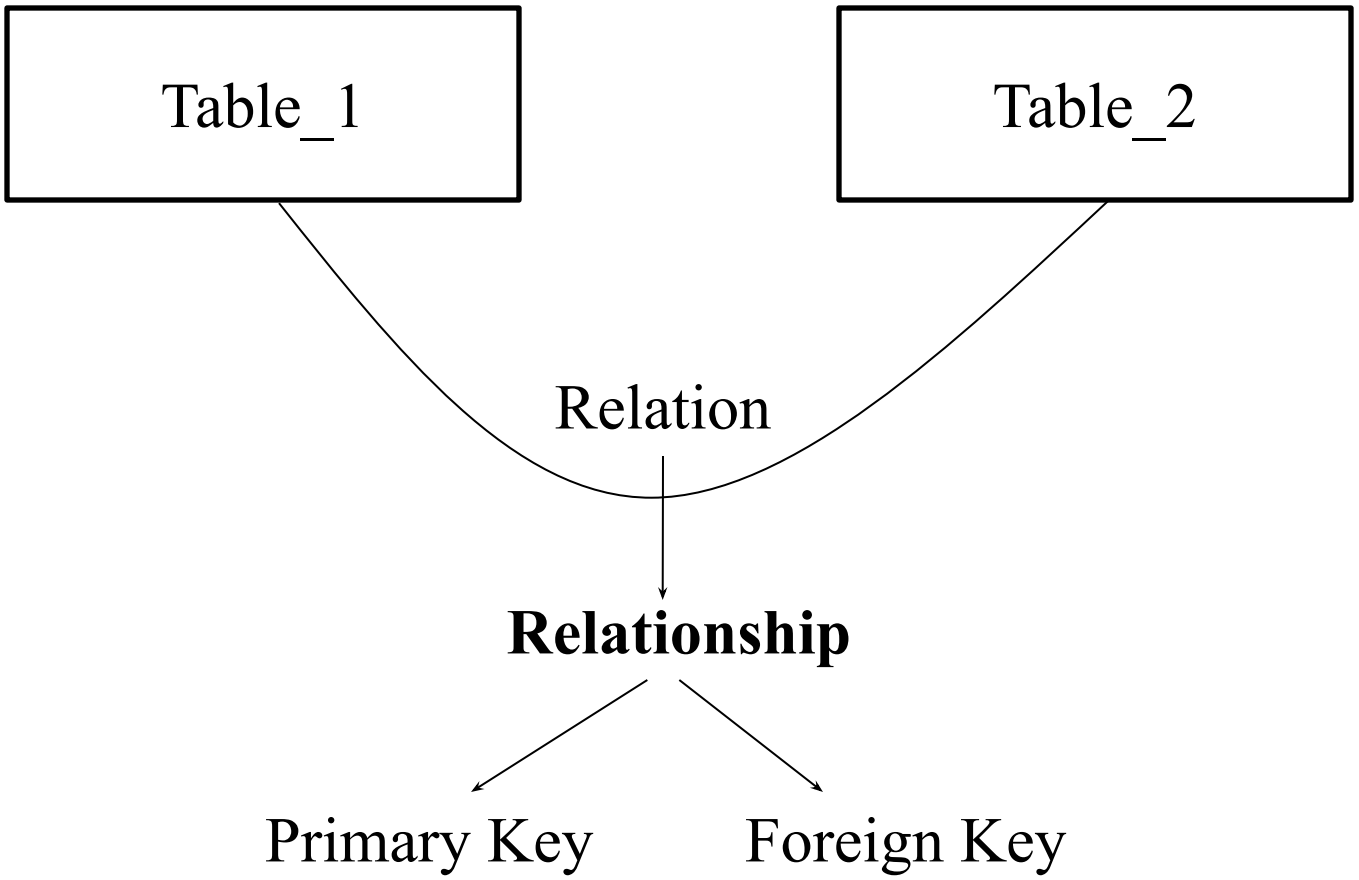
- Example: Age is derived attribute, because it is not physically stored in the database. it is calculated with the help of "DOB".



Example- ER Model



Relationship



Types of Relationship

- One-to-one
- One-to-many
- Many-to-many

Types of Relationship

- **One-to-one(1:1) Relationship:** A one-to-one (1:1) relationship is when at most one instance of a entity A is associated with one instance of entity B.(i.e. one record of table-1 is related to only one record of table-2 and vice versa)

One-to-one(1:1) Relationship

Table_1

Person			

Table_2

Passport			

Types of Relationship

- **One-to-many(1:M) Relationship:** A one-to-many (1:M) relationships is 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.

One-to-many(1:M) Relationship

Table_1

Company			

Table_2

Employee			

Types of Relationship

- **Many-to-many(M:M) Relationship:** A many-to-many (M:M) relationship, sometimes called non-specific, is 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.

Many-to-many(M:M) Relationship

Table_1

Student			

Table_2

Course			

Many-to-many(M:M) Relationship Example