**ER Model**

* Entity-Relationship model is used in the conceptual design of a database.
* Design is independent of all physical considerations (DBMS, OS).
* What does Conceptual Design include?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ideas −→ | High-level design | −→ | Relational database schema | −→ | Relational  DBMS |
|  |  |  |  |  |  |

* **Entity***: -* Real-world object or thing with an independent existence and which is distinguishable from other objects.

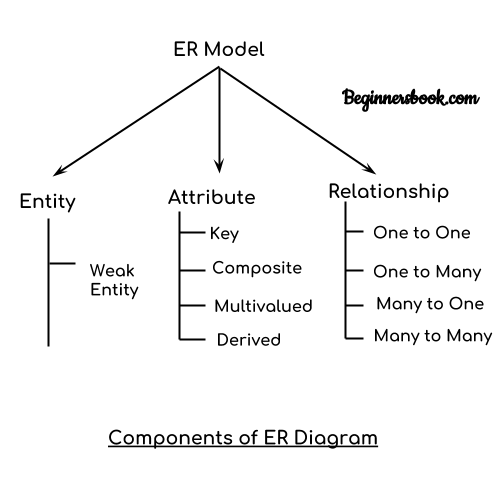
Examples: - a person, student, car, customer, product, gene, book etc.

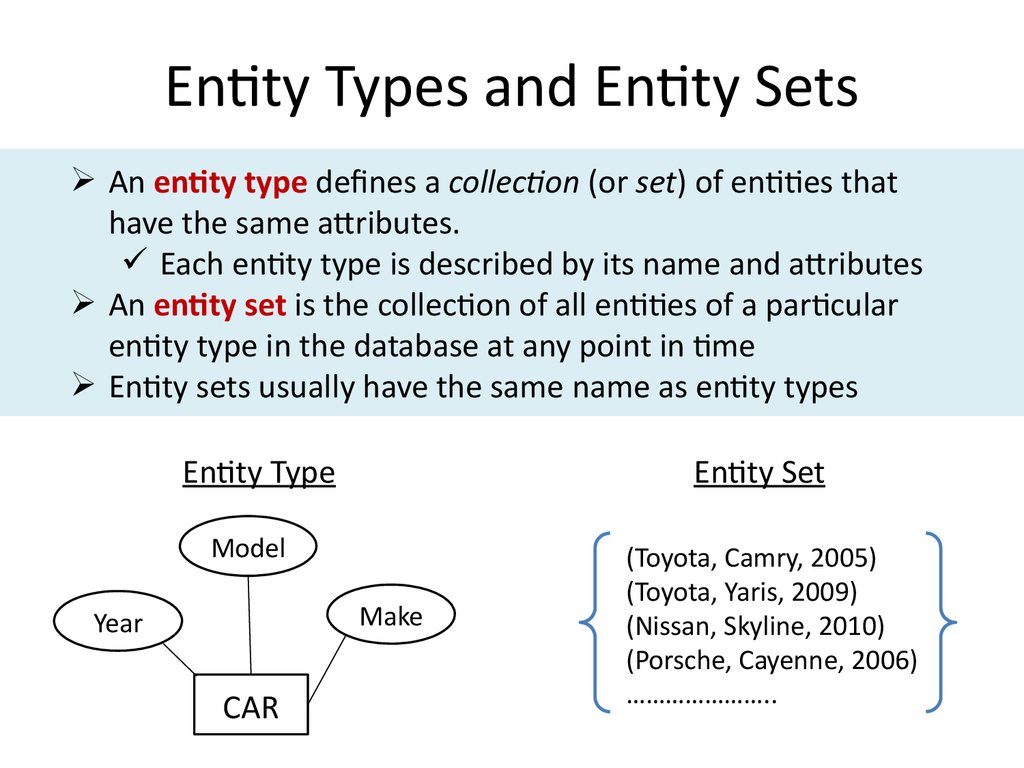
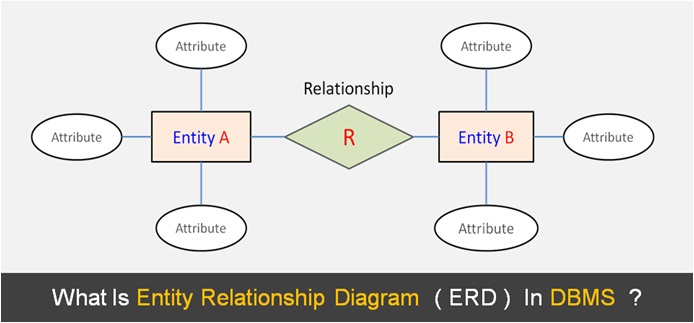
* **Attributes***: -* An entity is represented by a set of attributes (its descriptive properties).

e.g., name, age, salary, price etc.

Attribute values that describe each entity become a major part of the data eventually stored in a database.

* With each attribute a *domain* is associated, i.e., a set of permitted values for an attribute. Possible domains are integer, string, date, etc.



* 
* Relationship: - It is the association among entities.
* *Degree of a relationship* refers to the number of entity types that participate in the relationship type (binary, ternary, . . . ).
* Note that a relationship does not have key attributes! The identification of a particular relationship in a relationship set occurs through the keys of participating entities.

# Example of an Entity-Relationship Diagram

offers

Price

SAddress

Chain

CAddress

CUSTOMERS

orders

Account

PRODUCTS

SUPPLIERS

Quantity

FName

LName

Prodname

SName

Customers-Suppliers-Products Entity-Relationship Diagram

* Rectangles represent entity types
* Ellipses represent attributes
* Diamonds represent relationship types
* Lines link attributes to entity types and entity types to relationship types
* Primary key attributes are underlined
* Double Ellipses represent multi-valued attributes

Types of attributes

* Single and multivalued attribute
* Simple and composite attribute
* Stored and derived attribute
* Key and non-key attribute
* Required and optional attribute

Degree of relationship

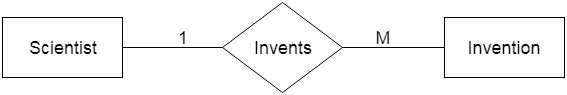
1. One to one relationship
2. One to many relationship
3. Many to one relationship
4. Many to many relationship
5. **One to one relationship**: - In one to one relationship, each record in one table is associated with one and only one record in another table.

Ex: - A female can marry to one male, and a male can marry to one female.



**b. One-to-many relationship**

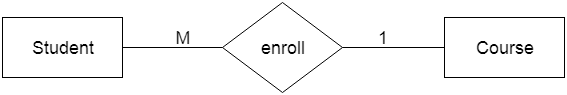
When only one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then this is known as a one-to-many relationship.

**Ex: -** Scientist can invent many inventions, but the invention is done by the only specific scientist.  


**c. Many-to-one relationship**

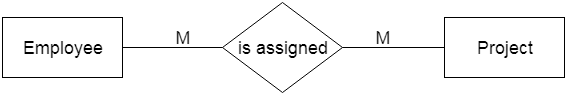
When more than one instance of the entity on the left, and only one instance of an entity on the right associates with the relationship then it is known as a many-to-one relationship.

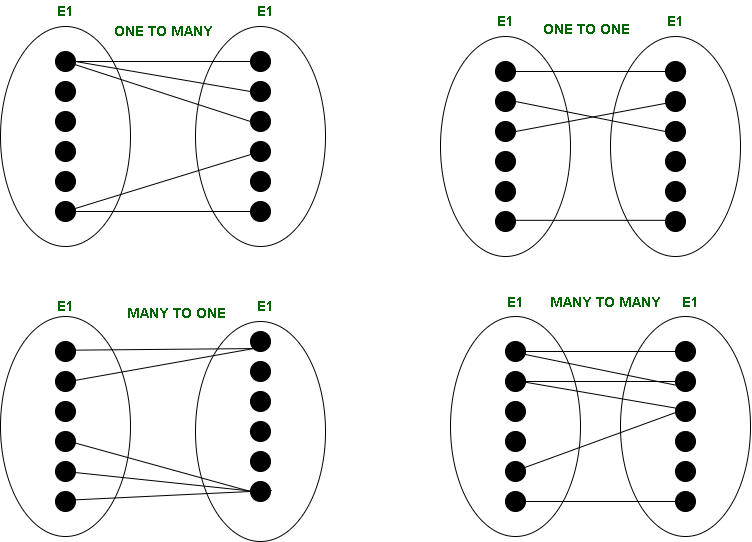
**For example,** Student enrolls for only one course, but a course can have many students.

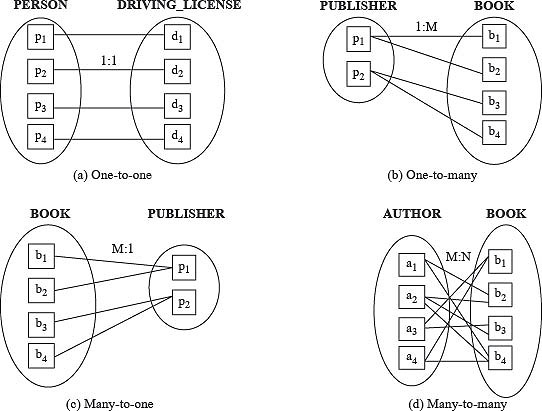


**d. Many-to-many relationship**

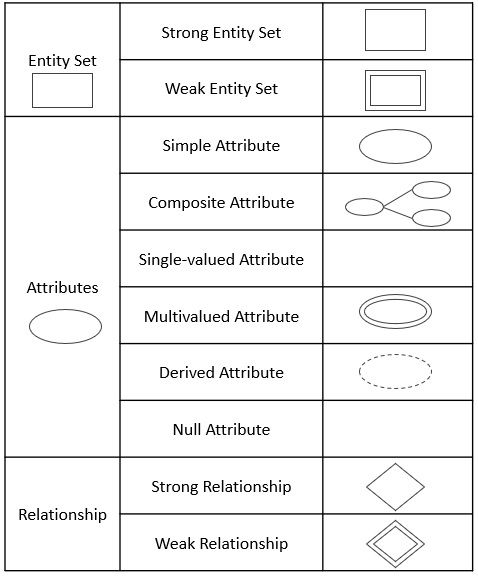
When more than one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then it is known as a many-to-many relationship.

**For example,** Employee can assign by many projects and project can have many employees.  




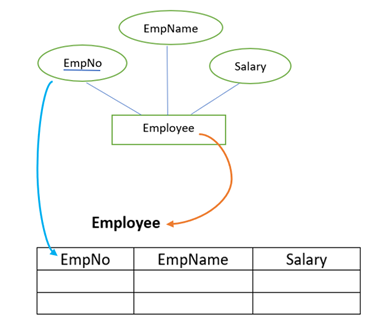


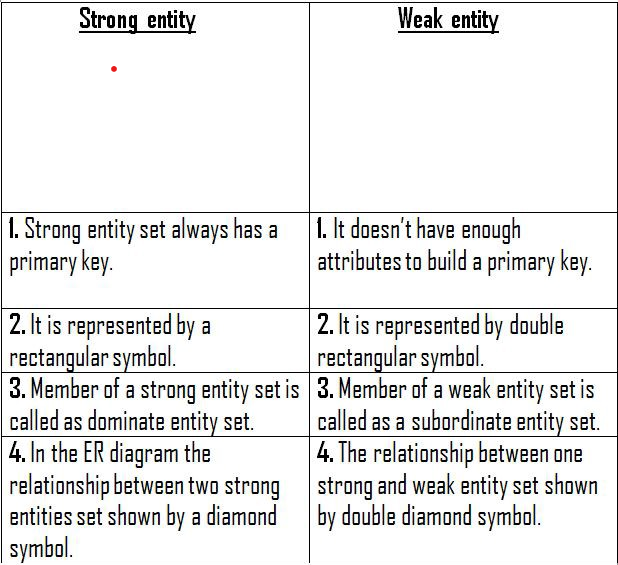
Symbols



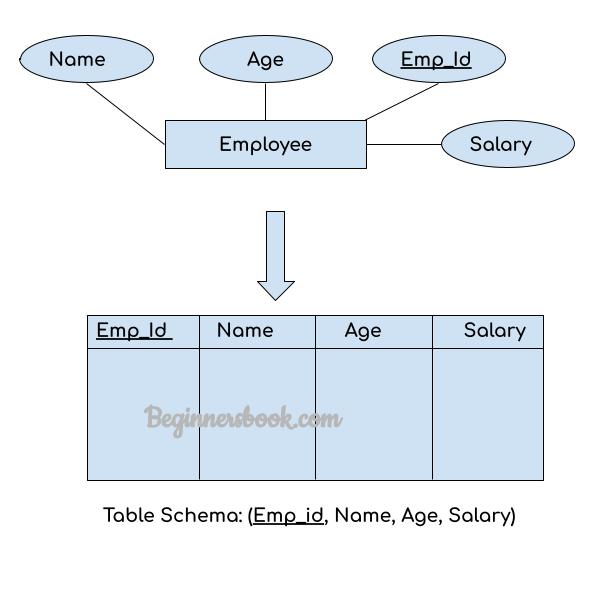
# Translation of ER Schema into Tables

* An ER schema can be represented by a collection of tables which represent contents of the database (instance).
* Primary keys allow entity types and relationship types to be expressed uniformly as tables.
* For each entity and relationship type, a unique table can be derived which is assigned the name of the corresponding entity or relationship type.
* Each table has a number of columns that correspond to the attributes and which have unique names. An attribute of a table has the same domain as the attribute in the ER schema.
* Translating an ER schema into a collection of tables is the basis for deriving a relational database schema from an ER diagram.

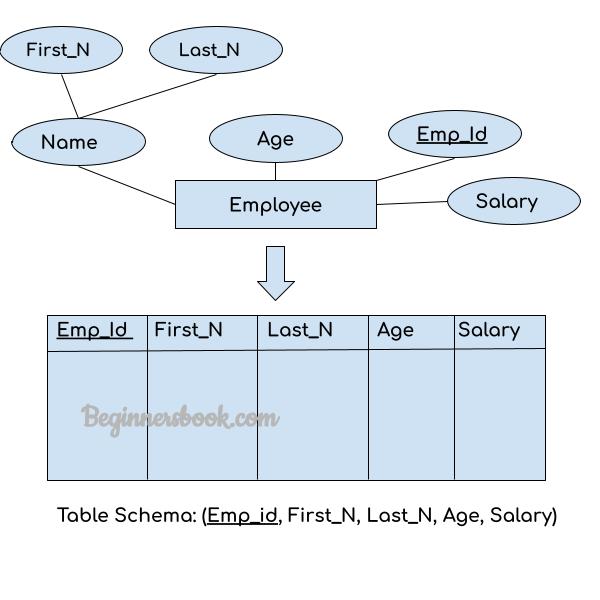




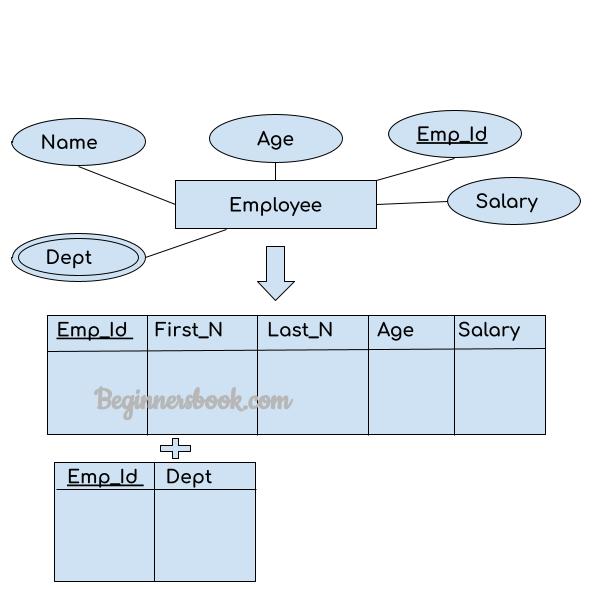
1. Strong entity set with simple attribute



1. Strong entity with composite attribute



1. Strong entity with multivalued attribute



# Enhanced ER Modeling Concepts

Although most properties of entities and relationships can be expressed using the basic modeling constructs, some of them are costly and difficult to express (and to understand). That’s why there are some extensions to the ER model.

# Subclasses, Superclasses, and Inheritance

* In some cases, an entity type has numerous subgroupings of its entities that are meaningful and need to be represented explicitly because of their significance to the DB application.

ISA

PERSON

GPA

STUDENT

Major

Name

SSN

* Relationships and attributes of superclass are inherited to subclass (in particular primary key attribute(s)); subclass can have additional attributes and relationships
* An entity cannot exist merely by being a member of only a subclass.

# Specialization

* Process of defining a set of subclasses of an entity type (top-down)

ISA

EMPLOYEES

SocialSN

HOURLY\_EMPS

Address

Name

CONTRACT\_EMPS

Contractno

Hours

Wages

HOURLY EMPS is a subclass of EMPLOYEES and thus inherits its attributes and relationships (same for CONTRACT EMPS).

Generalization:

* Reverse process of specialization (bottom-up); identify common features of entity types and generalize them into

single superclass (including primary key!)

ISA

CAR

TRUCK

Tonage

Axels

MaxSpeed

NoOfPassengers

VEHICLE

VehicleNo

Price

LicensePlate

-------THANK YOU-------

(By: - Shivangi)