



# S. S Jain Subodh P.G. (Autonomous) College

SUBJECT - DATABASE MANAGEMENT SYSTEM

TITLE – E-R MODEL IN DBMS

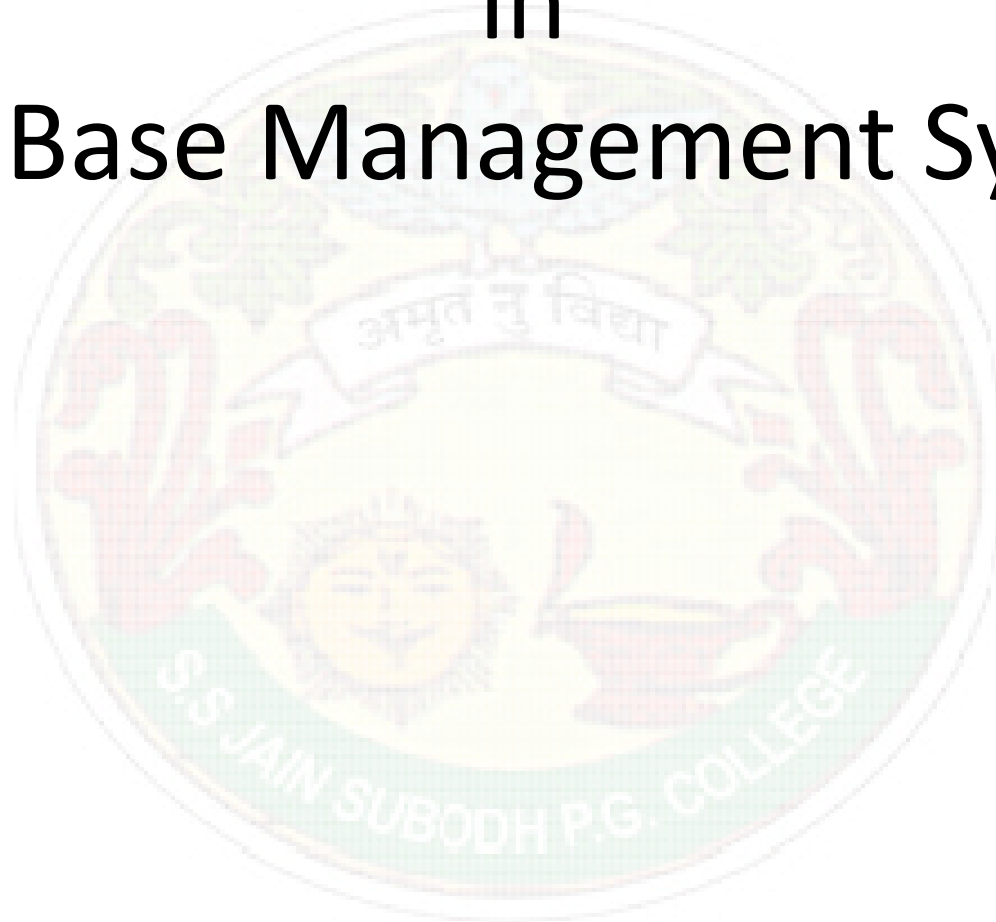


Presented By  
**Vandana Nigam**



S. S Jain Subodh P.G. (Autonomous) College

# A Presentation on E-R Models in Data Base Management System





## E-R model

Entity-Relationship (ER) Model is based on the notion of real-world entities and relationships among them. While formulating real-world scenario into the database model, the ER Model creates entity set, relationship set, general attributes and constraints.

- ER Model is best used for the conceptual design of a database.
- ER Model is based on –
- **Entities** and their *attributes*.
- **Relationships** among entities.

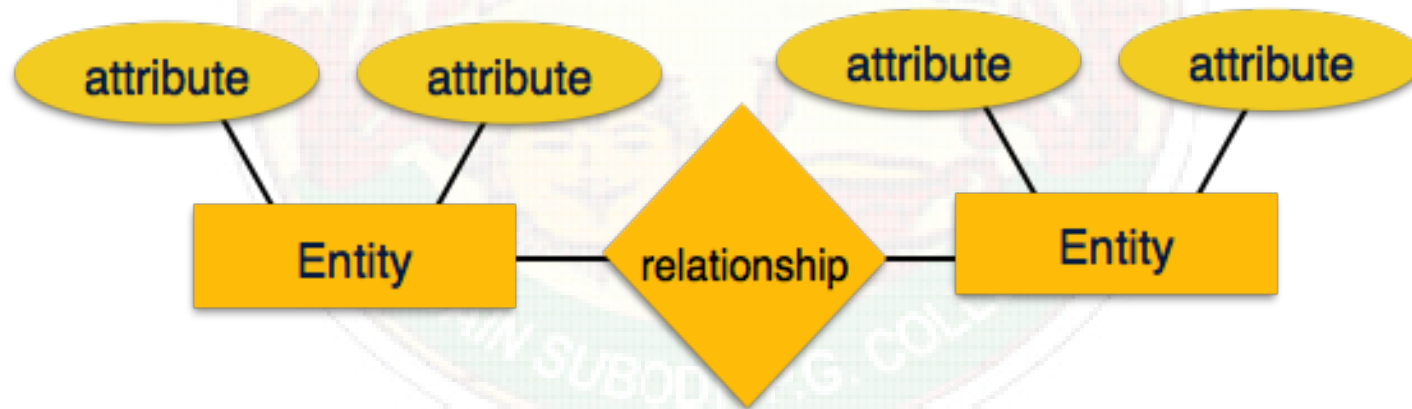


# E-R model

ER Model is based on –

**Entities** and their *attributes*.

**Relationships** among entities.





# Entity

An entity can be a real-world object, either animate or inanimate, that can be easily identifiable. For example, in a school database, students, teachers, classes, and courses offered can be considered as entities. All these entities have some attributes or properties that give them their identity. An entity set is a collection of similar types of entities. An entity set may contain entities with attribute sharing similar values. For example, a Students set may contain all the students of a school; likewise a Teachers set may contain all the teachers of a school from all faculties. Entity sets need not be disjoint.



# Attributes

Entities are represented by means of their properties called attributes. All attributes have values. For example, a student entity may have name, class, and age as attributes. There exists a domain or range of values that can be assigned to attributes. For example, a student's name cannot be a numeric value. It has to be alphabetic. A student's age cannot be negative, etc.

## Types of Attributes

- **Simple attribute:** Simple attributes are atomic values, which cannot be divided further. For example, a student's phone number is an atomic value of 10 digits.
- **Composite attribute:** Composite attributes are made of more than one simple attribute. For example, a student's complete name may have first\_name and last\_name.
- **Derived attribute:** Derived attributes are the attributes that do not exist in the physical database, but their values are derived from other attributes present in the database. For example, average\_salary in a department should not be saved directly in the database, instead it can be derived. For another example, age can be derived from data\_of\_birth.



## Types of Attributes

**Simple attribute:** Simple attributes are atomic values, which cannot be divided further. For example, a student's phone number is an atomic value of 10 digits.

**Composite attribute:** Composite attributes are made of more than one simple attribute. For example, a student's complete name may have first\_name and last\_name.

**Derived attribute:** Derived attributes are the attributes that do not exist in the physical database, but their values are derived from other attributes present in the database. For example, average\_salary in a department should not be saved directly in the database, instead it can be derived. For another example, age can be derived from data\_of\_birth.



# Types of Attributes Codd.

**Single-value attribute:** Single-value attributes contain single value. For example: Social\_Security\_Number.

**Multi-value attribute:** Multi-value attributes may contain more than one values. For example, a person can have more than one phone number, email\_address, etc.

These attribute types can come together in a way like:

- simple single-valued attributes
- simple multi-valued attributes
- composite single-valued attributes
- composite multi-valued attributes





# Entity-Set and Keys

Key is an attribute or collection of attributes that uniquely identifies an entity among entity set. For example, the roll\_number of a student makes him/her identifiable among students.

- **Super Key:** A set of attributes (one or more) that collectively identifies an entity in an entity set.
- **Candidate Key:** A minimal super key is called a candidate key. An entity set may have more than one candidate key.
- **Primary Key:** A primary key is one of the candidate keys chosen by the database designer to uniquely identify the entity set.



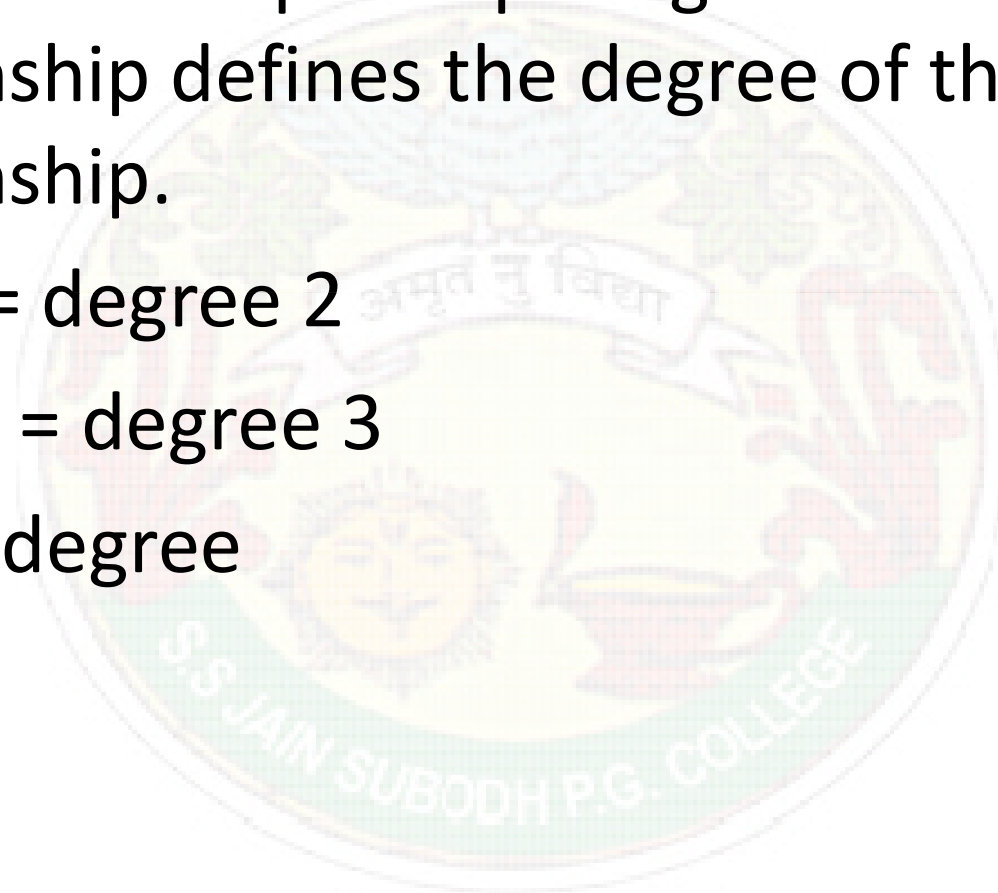
# Relationship

- **Relationship** – The logical association among entities is called *relationship*.
- Relationships are mapped with entities in various ways. Mapping cardinalities define the number of association between two entities.
- Mapping cardinalities –
  - one to one
  - one to many
  - many to one
  - many to many



# Degree of Relationship

- The number of participating entities in a relationship defines the degree of the relationship.
- Binary = degree 2
- Ternary = degree 3
- n-ary = degree

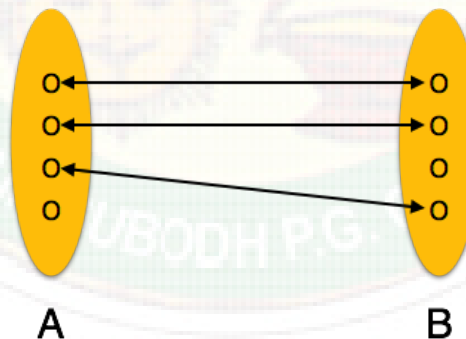




## Mapping Cardinalities

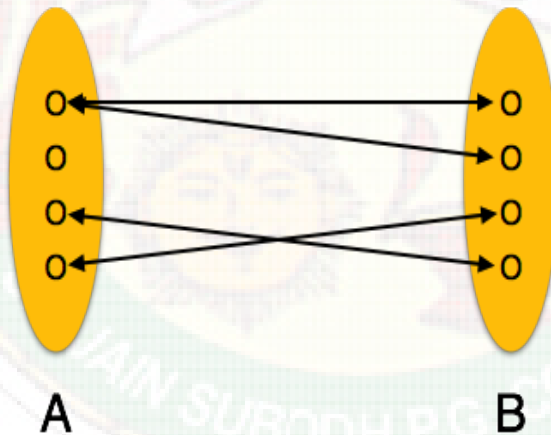
**Cardinality** defines the number of entities in one entity set, which can be associated with the number of entities of other set via relationship set.

- **One-to-one** – One entity from entity set A can be associated with at most one entity of entity set B and vice versa.





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





- **Many-to-many** – One entity from A can be associated with more than one entity from B and vice versa.

