

# **BCA SEMESTER-IV**

## **Database Management Systems**

**Paper Code: BCA CC410**

**Unit :2**

### **Conceptual Modeling**

**By:**

**Ms. Sushmita Chakraborty**

Assistant Professor

Dept. of MCA

Patna Women's College

# Database Management Systems

## Unit : 2 Conceptual Modelling

The conceptual model can be represented using two major approaches namely Entity-Relationship Modelling and Object Modeling.

### E-R Model & its advantages

It is the most popular conceptual model used for designing a database . The E-R model views the real world as a set of basic objects ( known as entities), their characteristics ( known as attributes ) and associations among these objects ( known as relationships).

The E-R model has several advantages as listed below:

1. It is simple and easy to understand and thus, can be used as an effective communication tool between the database designer and the end user.
2. It captures the real-world data requirements in a simple, meaningful and logical way.
3. It can be easily mapped to the relational model. The basic constructs that is the entities and attributes of the E-R model can be easily transformed into relations and columns.
4. It can be used as a design plan and can be implemented in any database management software.

**Entity :** It is a distinguishable object that has an independent existence in the real world. An entity can exist either physically or conceptually.

- **Tangible entity :** If an entity has a physical existence , it is termed as tangible entity or concrete entity.
- **Non-tangible :** If an entity has a conceptual existence, it is termed non-tangible or abstract entity.

- **Entity type** : A set or a collection of entities that share the same attributes but different values is known as an entity type.
- **Instance** : A specific occurrence of an entity type is called its instance. Eg. The publisher *Hills Publication* is the instance of *entity type* PUBLISHER.
- **Schema**: Each entity is referred to by its name and attribute values. An entity type describes the schema for the entity set that share the same structure.

**Attribute** : Attributes are the properties of an entity that characterize and describe it. Each attribute can accept a value from a set of permitted values which is called the domain or the value set of the attribute.

### Types of attributes

**B1----- Book\_title**

**Price=22**

**ISBN=001-354-921—1**

**Year=2020**

**Page\_count**

**Category="Novel"**

**P1----- P\_ID**

**Name=Hills Publication**

**Address= gjhd**

**Phone = 689797**

**Email-id=jdhc@gmail.com**

- **Identifying and descriptive attributes** :The attribute that is used to uniquely identify an instance of an entity is known as

**Identifying and descriptive attributes or simple an identifier.**

**Eg:** the attribute P\_ID of the entity type PUBLISHER is an identify attributes. As two publishers cannot have the same publisher ID. A descriptive attribute or simply a descriptor on the other hand describe a non-unique characteristics of an entity instance. Eg: the attribute *Price & Page\_count* are the descriptive attributes as two books can have the same price and number of pages.

- **Simple & composite attributes :** The attributes that are indivisible are known as simple attributes. Eg : the attributes Book\_title, Price, Year, page\_count and category of the entity BOOK are simple attributes as they cannot be further divided into smaller subparts.

**Composite:** can be divided in to smaller subparts eg: the attribute *Address* can be further divided into *House\_number*  
*Street, City, State and Zip\_code*.

- **Stored & derived attributes:** Suppose an entity type PERSON nad its attributes *Date\_of\_birth* and *Age*. The value of the attribute *Age* can be determined from the current date and the value of *Date\_of\_birth*. Thus, the attribute *Age* is known as **derived attribute** and the attribute *Date\_of\_birth*. is known as a stored attribute.
- **Single-valued and multi-valued:** The attribute that can have only one value for a given entity are called the single-valued attributes. Eg: the attribute BOOK\_titlt is a single-valued attribute as on book can have only one title. The attribute that can have multiple values for a given entity are called multi-valued attributes. Eg: email-id & phone attribute of the entity type PUBLISER are multi-valued attributes as publisher can have zero, one or more e-mail IDs and phone numbers.

- **Complex attributes:** The attributes that are formed by arbitrarily nesting the composite and multi-valued attributes are called **Composite attributes**. This is represented by grouping the components of a composite attribute between parenthesis () and by displaying the multi-valued attributes between braces {} and attributes separated by commas.

### **Various situations in which an attribute can use a null value**

There are various situations in which an attribute can use a NULL value. One such situation is when a particular entity does not have an appropriate value for an attribute. In such situation the attribute take a special value called the NULL value. A null value of an attribute may indicate “not applicable”. Eg: consider an entity type PERSON with three attributes First\_name, Middle\_name and Last\_name. Since not every person has a middle name, the person without any middle name will have a NULL value for the attribute Middle\_name.

### **Strong Entity and Weak Entity**

- An entity type that does not have any key attribute of its own is called a weak entity type.
- Whereas an entity type that has a key attribute is called a strong entity type.

The weak entity is also called a dependent entity as it depends on another entity for its identification. The strong entity is called an independent entity as it does not rely on another entity for its identification. The weak entity has no relevance and meaning in an E-

R diagram. Eg: an entity type EDITION with its attributes Edition\_no and Type. It depends on another entity type BOOK for its existence, as an edition cannot exist without a book. Thus EDITION is a weak entity type and BOOK is a strong entity type.