

UNIT 1

Lecture 5

E R Model

E R Model

- The **entity-relationship (E-R)** data model perceives the real world as consisting of basic objects, called ***entities***, and ***relationships*** among these objects.
- It was developed to facilitate database design by allowing specification of an ***enterprise schema***, which represents the overall logical structure of a database.
- The E-R model is very useful in mapping the meanings and interactions of real-world enterprises onto a conceptual schema.

Entity

- An **entity** is a “thing” or “object” in the real world that is distinguishable from all other objects.
- For example, each person in an enterprise is an entity.
- An entity has a set of properties, and the values for some set of properties may uniquely identify an entity.
- For instance, a person may have a *person-id* property whose value uniquely identifies that person. Thus, the value 677-89-9011 for *person-id* would uniquely identify one particular person in the enterprise.

Entity Set

- An **entity set** is a set of entities of the same type that share the same properties, or attributes.
- For E.g., The set of all persons who are customers at a given bank can be defined as the entity set ***customer***.
- Examples :
 - Person: PROFESSOR, STUDENT
 - Place: STORE, UNIVERSITY
 - Object: MACHINE, BUILDING
 - Event: SALE, REGISTRATION
 - Concept: ACCOUNT, COURSE

Entity Set Representation

STUDENT

COURSE

PERSON

CUSTOMER

TEACHER

BOOK

ACCOUNT

Entity Set Naming Conventions

Entity set name should be :

- A ***singular noun*** and in ***capital letters***.
- ***Descriptive*** and ***specific*** to the organization.
- ***Concise***.
- Named for the ***result of the event***, not the activity or process of the event.

Extension

- The individual entities that constitute a set are said to be the ***extension*** of the entity set.
- Thus, all the individual bank customers are the extension of the entity set ***customer***.

Attribute

- An entity is represented by a set of **attributes**.
- Attributes are descriptive properties possessed by each member of an entity set.
- The designation of an attribute for an entity set expresses that the database stores similar information concerning each entity in the entity set; however, each entity may have its own value for each attribute.
- Possible attributes of the ***customer*** entity set are *customer-id*, *customer-name*, *customer-street*, and *customer-city*.

Domain (Value Set)

- For each attribute, there is a set of permitted values, called the **domain**, or **value set**, of that attribute.
- The domain of attribute *customer-name* might be the set of all text strings of a certain length.

Attribute (Naming Guidelines)

- **An attribute name:**
 - Should be a *noun* and *capitalize the first letter of each word*.
 - Example: Student_ID
 - Should be *unique*.
 - Should follow a *standard format*.
 - Example: Student_GPA, not GPA_of_Student
- Similar attributes of different entity types should use similar but distinguished names.
 - Example: Faculty_Residence_City_Name and Student_Residence_City_Name

Attributes

- STUDENT
 - Rollno, Sname, Branch, Sem, Address, Dob, Age, Phone_No
- TEACHER
 - Tid, Tname, Specialization
- COURSE
 - Course_Id, Course_Name, Syllabus, Duration, Credits
- BOOK
 - Book_Id, Title, Author, Price, No_Of_Pages
- PERSON
 - Person_Id, Pname, Address
- ACCOUNT
 - Acc_No, Balance
- CUSTOMER
 - Cust_No, Cust_Name, Address

Types of attributes in the ER Model

- The attributes used in the ER model can be categorized as
 - Simple or Composite
 - Single Valued or Multi Valued
 - Stored or Derived
 - NULL or prime

Simple or Composite

- The attribute which are not divided into subparts are called **simple** attributes.
 - For example, an attribute *customer-id* is a simple attribute.
- **Composite** attributes, on the other hand, can be divided into subparts (that is, other attributes).
 - For example, an attribute *name* could be structured as a composite attribute consisting of *first-name*, *middle-initial*, and *last-name*.
- Using composite attributes in a design schema is a good choice if a user will wish to refer to an entire attribute on some occasions and to only a component of the attribute on other occasions.
- Suppose we were to substitute for the *customer* entity-set attributes *customer-street* and *customer-city* the composite attribute *address* with the attributes, *street*, *city*, *state*, and *zip-code*.
- Note also that a composite attribute may appear as a hierarchy. In the composite attribute *address*, its component attribute *street* can be further divided into *street-number*, *street-name*, and *apartment-number*.

Single Valued or Multi Valued

- The attributes that have a single value for a particular entity is called a **single valued** attribute.
- For example, an attribute *customer-id* is a single valued attribute because for a particular entity it holds a single value.
- The attribute that have multiple valued for a particular entity is called a **multi valued** attribute.
- For example, an attribute phone-number is a multi valued attribute because for a particular customer it holds zero, one or several phone numbers.

Stored or Derived

- Normally attributes are **stored** attributes, that is, their values are stored and accessed as such from the database.
- For example, the attributes *name*, *address* and *date-of-birth* of customer entity set are stored attributes.
- However, sometimes attributes' values are not stored as such, rather they are computed or derived based on some other value. This other value may be stored in the database or obtained some other way.
- For example, we may store the *name*, *father-name*, *address* of customers, but *age* can be computed from *date-of-birth*.
- The advantage of declaring age as derived attribute is that whenever we will access the age, we will get the accurate, current age of employee since it will be computed right at the time when it is being accessed.

Null or Prime

- An attribute takes a **null** value when an entity does not have a value for it.
- The ***null*** value may indicate “not applicable”—that is, that the value does not exist for the entity.
- For example, one may have no middle name.
- ***Null*** can also designate that an attribute value is unknown. An unknown value may be either ***missing*** (the value does exist, but we do not have that information) or ***not known*** (we do not know whether or not the value actually exists).
- An attribute whose values are used to identify an entity within an entity set is known as **prime** attribute.

Symbols for Attributes

Simple



Composite

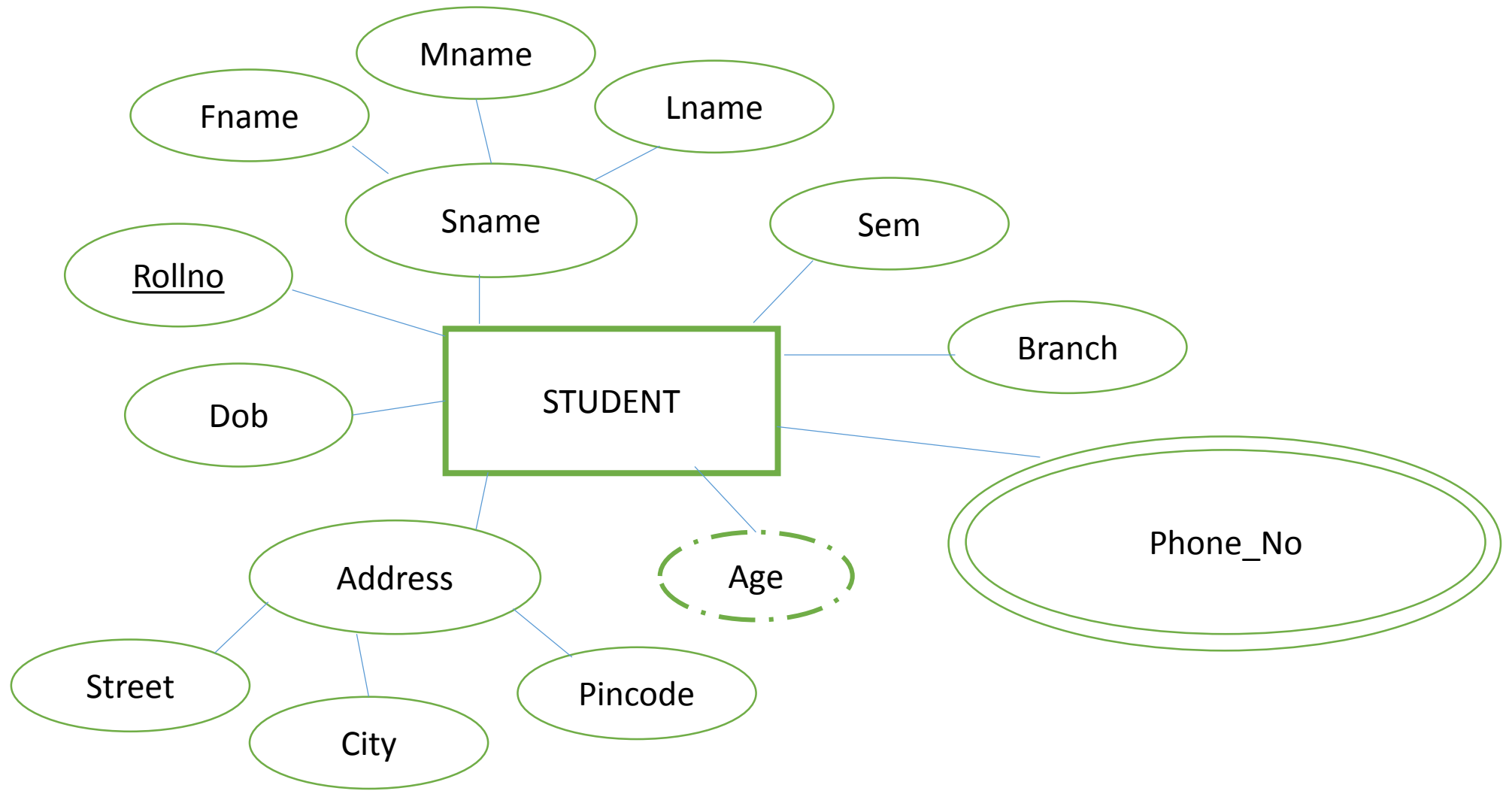


Multi-valued



Derived





University Questions

- What are the types of attributes used in the Entity Relationship Model?

For Video lecture on this topic please subscribe to my youtube channel.

The link for my youtube channel is

https://www.youtube.com/channel/UCRWGtE76JITp1iim6aOTRuW?sub_confirmation=1