Entity-Relationship (E/R) Model

Dr P Sreenivasa Kumar
Professor
CS&E Dept I I T Madras

Entity-Relationship (E/R) Model

- Widely used conceptual level *data model*
  - proposed by Peter P Chen in 1970s
- Data model to describe the database system at the requirements collection stage
  - high level description.
  - easy to understand for the enterprise managers.
  - rigorous enough to be used for system building.
- Concepts available in the model
  - entities and attributes of entities.
  - relationships between entities.
  - diagrammatic notation.

Entities

- *Entity* - a thing (animate or inanimate) of independent physical or conceptual existence and *distinguishable*.

  In the University database context, an individual *student*, *faculty member*, a *class room*, a *course* are entities.

- *Entity Set* or *Entity Type*-

  Collection of entities all having the same properties.
  Student entity set – collection of all *student* entities.
  Course entity set – collection of all *course* entities.

Attributes

Each entity is described by a set of attributes/properties
that have associated values

student entity

- *StudName* – name of the student.
- *RollNumber* – the roll number of the student.
- *Sex* – the gender of the student etc.

All entities in an Entity set/type have the same set of attributes.

Chosen set of attributes – amount of detail in modeling.

Types of Attributes (1/2)

- Simple Attributes
  - having atomic or indivisible values.

    example: *Dept* – a string
    PhoneNumber – a ten digit number
- Composite Attributes
  - having several components in the value.

    example: *Qualification* with components
    (*DegreeName*, *Year*, *UniversityName*)
- Derived Attributes
  - Attribute value is dependent on some other attribute.

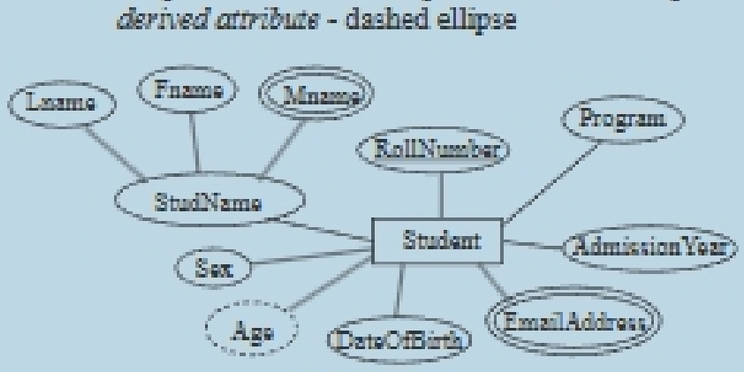
    example: *Age* depends on *DateOfBirth*.
    So age is a derived attribute.

Types of Attributes (2/2)

- Single-valued
  - having only one value rather than a set of values.
  - for instance, *PlaceOfBirth* – single string value.
- Multi-valued
  - having a set of values rather than a single value.
  - for instance, *CoursesEnrolled* attribute for student
    EmailAddress attribute for student
    PreviousDegree attribute for student.
- Attributes can be:
  - simple single-valued, simple multi-valued,
  - composite single-valued or composite multi-valued.

Diagrammatic Notation for Entities

entity - rectangle
attribute - ellipse connected to rectangle
multi-valued attribute - double ellipse
composite attribute - ellipse connected to ellipse
derived attribute - dashed ellipse



Domains of Attributes

Each attribute takes values from a set called its *domain*
For instance, *studentAge* – {17,18, …, 55}
HomeAddress – character strings of length 35
Domain of composite attributes –
cross product of domains of component attributes
Domain of multi-valued attributes –
set of subsets of values from the basic domain

Entity Sets and Key Attributes

- *Key* – an attribute or a collection of attributes whose value(s) uniquely identify an entity in the entity set.
- For instance,
  - *RollNumber* - Key for *Student* entity set
  - *EmpID* - Key for *Faculty* entity set
  - *HostelName, RoomNo* - Key for *Student* entity set (assuming that each student gets to stay in a single room)
- A key for an entity set may have more than one attribute.
- An entity set may have more than one key.
- Keys can be determined only from the meaning of the attributes in the entity type.
  - Determined by the designers