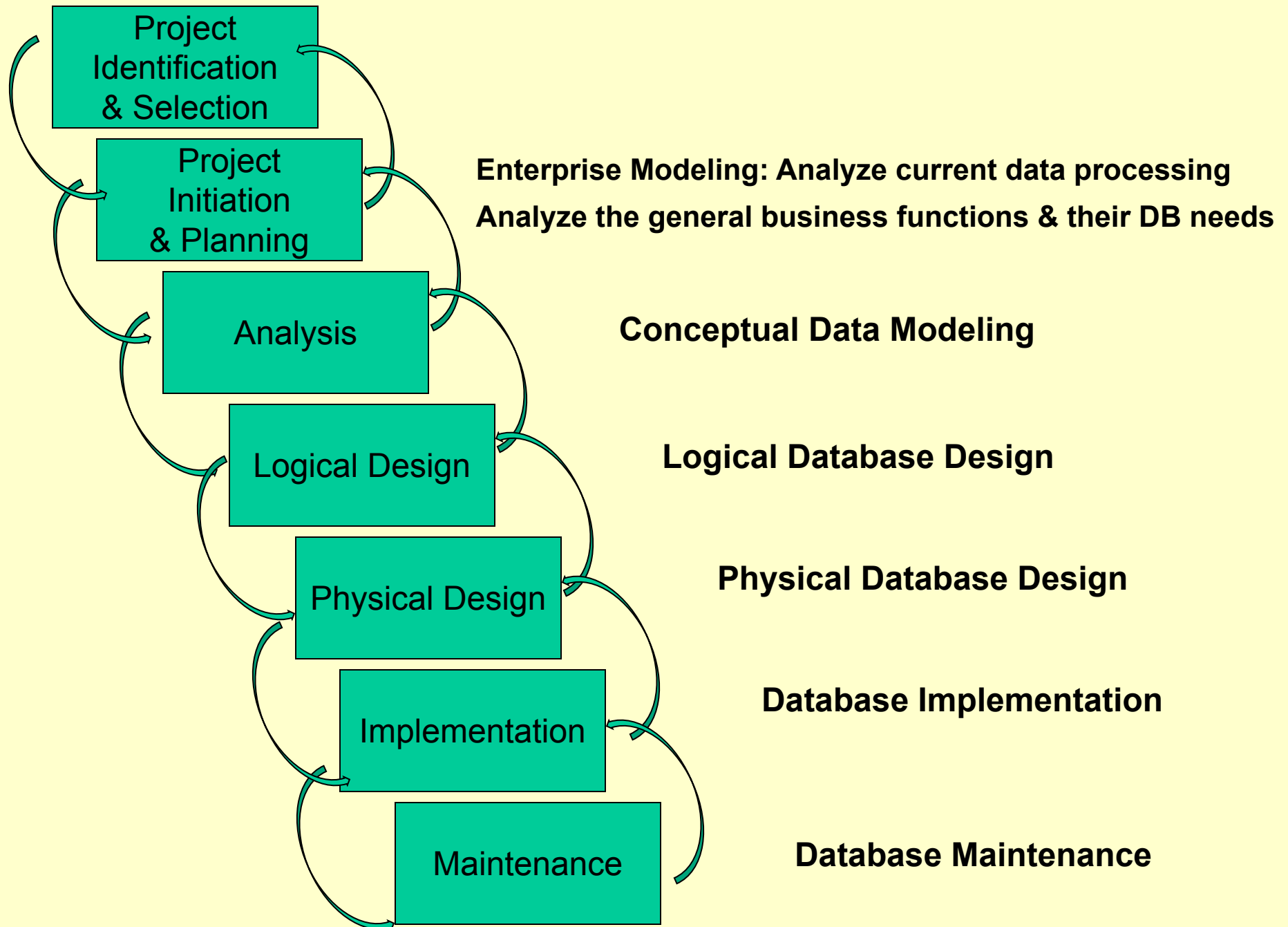


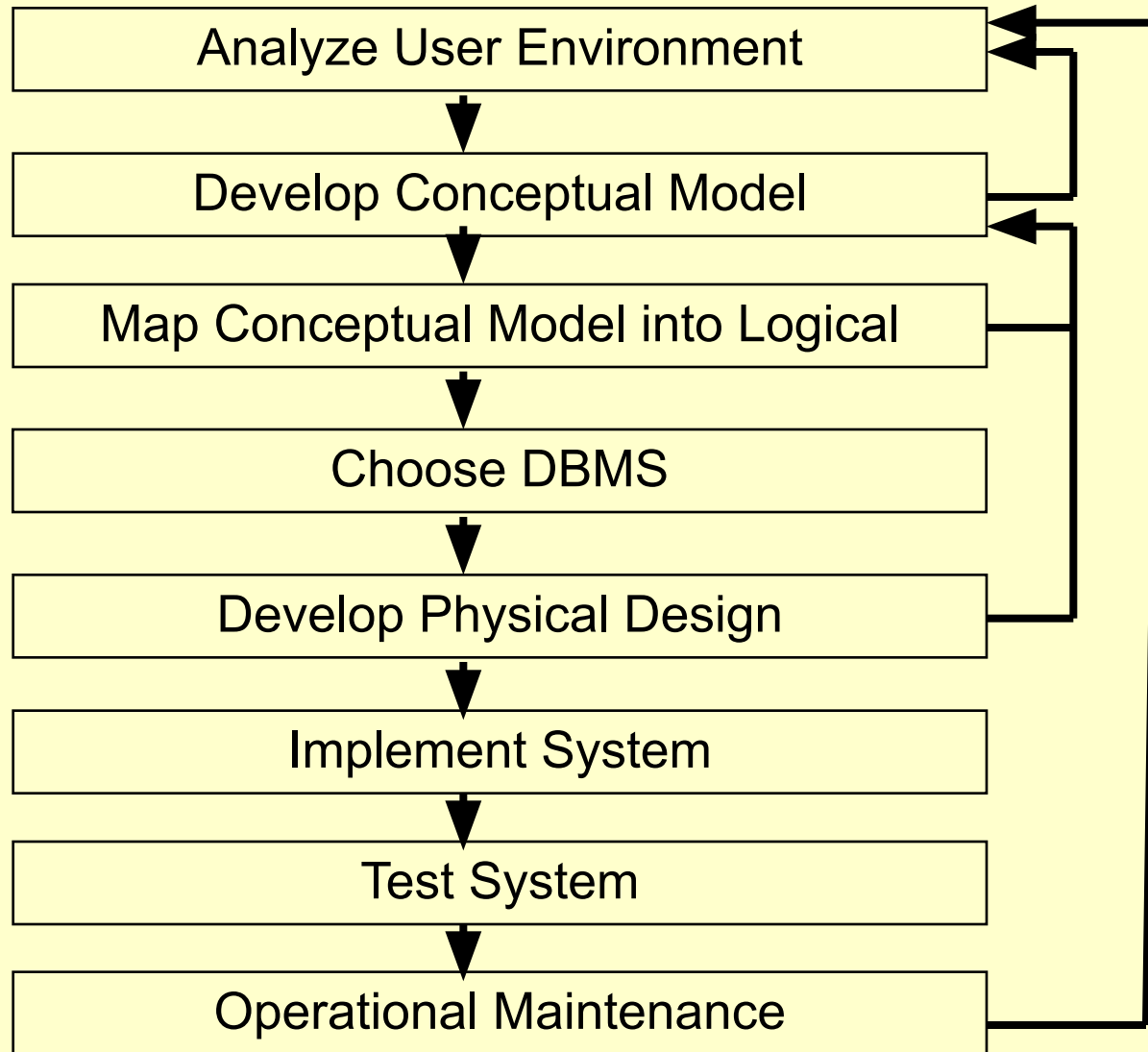
Database Systems



Database Development Life Cycle

- It is important to recognize that the stages of the database development life cycle are not strictly sequential, but involve some of amount of repetition of previous stages through feedback loops.
- For example, problems encountered during database design may necessitate additional requirements collection and analysis.

Database Development Life Cycle



System Models

- Webster's Dictionary defines model as a “description or analogy used to visualize something that cant be directly observed”.
- Models are abstraction of a more complex real-world object or event.
- The System models are the graphical representation of a System.
- Models are more understandable than detailed natural language description of the system.
- Models are an abstraction of the system. It leaves out details.
- Models are helpful for:
 - Testing a physical entity before building it
 - Communication with the customer
 - Visualization
 - Reduction of Complexity
 - Concept building and understanding

Data Modeling

- A data model is an integrated collection of concepts for describing and manipulating data, relationships between data, and constraints on the data in an organization.
- It is the first step in designing a database.
- It is the process of creating the logical structure of the database.
- Two main purposes of data modeling are to assist in the understanding of the data and to facilitate communication about information requirements.
- Building a data model requires answering questions about **entities, attributes, relationships and constraints**. In doing so, the designer discovers the semantics of the enterprise's data.
- Data Models are governed by the business rules of an organization.
- It is unlikely to create a good database without first selecting an appropriate data model.

Data Modeling

- A data model has the following three components:
 1. **Structural Part:** How the data are structured?
 2. **Manipulative Part:** Defines the types of operations that are allowed on the data. This includes the operations that are used for updating or retrieving data and for changing the structure of the database.
 3. **Integrity Constraints.** Defines the checks that are used to ensures correctness of data.

Data Modeling

- In Database, we come across the following types of models:

1. Conceptual Models

E-R Model

2. Logical Data Models

a) Hierarchical Data Model

b) Network Data Model

c) Relational Data Model

- The basic building blocks of all the data models are **entities, attributes, relationships and constraints** that are determined on the basis of **business rules** of an organization.
- A **business rule** is a brief and precise description of policy or principle of an organization.

Entity

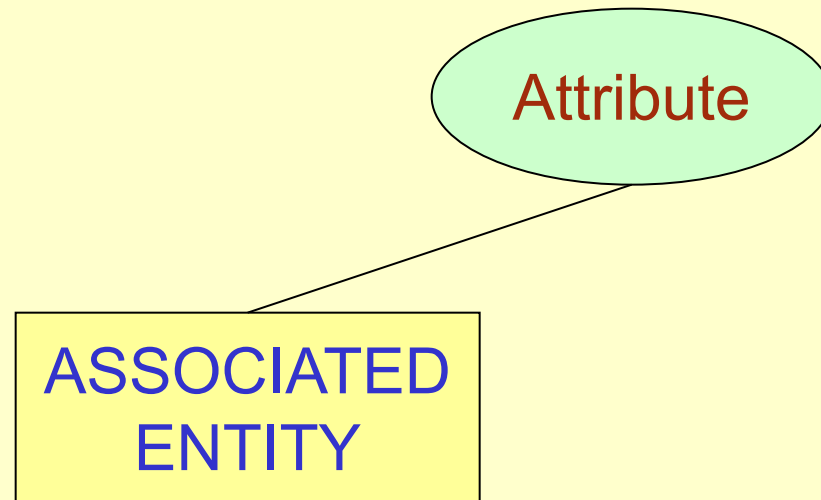
- An Entity is an **Persons, Place, Object, Event or Concept** about which the organization wishes to maintain data.
- An **Entity Type** or **Entity Class** is a collection of entities that can be described under the same set of characteristics.
- An **Entity Instance** is a single occurrence of an entity type.
- Generally, an entity in the real world is defined under a single entity type.
- An Entity is diagrammatically represented as



STUDENT

Attributes

- An Attribute is a property or characteristics of an entity type that is of interest to the organization.
- The attributes of a entity are selected on the basis of information requirements of an organization.
- For Ex, the attribute of a product for Sales Department will be different than that for Manufacturing Department.



Attributes

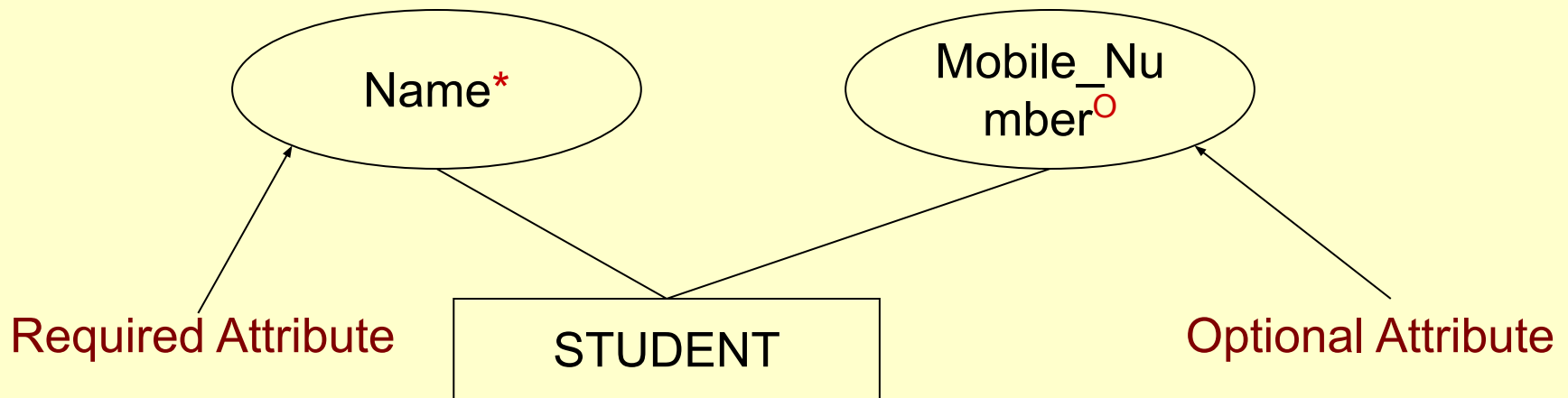
- Each entity in an entity class must possess at least one attribute that distinguishes it from other entities in the same entity class.
- This unique attribute of an entity is called an **Identifier**.
- **Domain** of an attribute is a set of permitted values for an attribute OR Domain is a pool of value from which an attribute extracts its actual value.
- **Null Value** of an attribute means the absence of value.
- Some less important attributes of an entity may be allowed to accept null values. Such attributes are called **optional attributes**.

Types of Attributes

- a) Required Attribute / Optional Attribute
- b) Atomic Attribute / Composite Attribute
- c) Single Valued Attribute / Multi valued Attribute
- d) Base Attribute / Derived Attribute

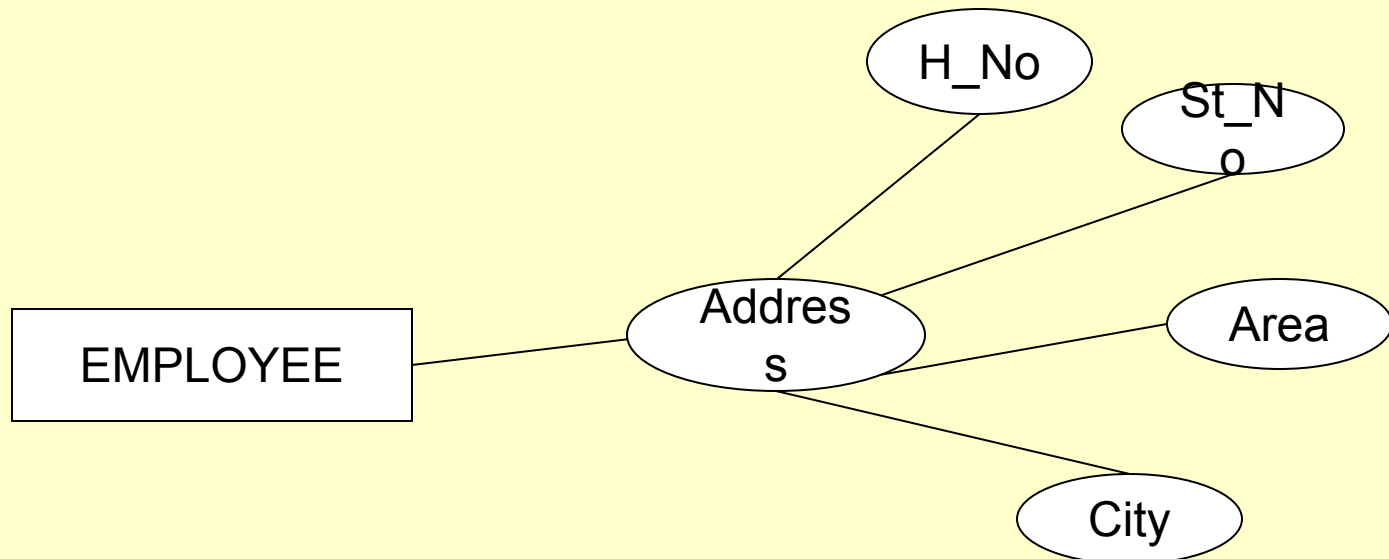
Types of Attributes

- **Required Attribute:** An attribute of an entity that must have a value for each entity instance
- **Optional Attribute:** An attribute of an entity that may/may not have a value for every entity instance.



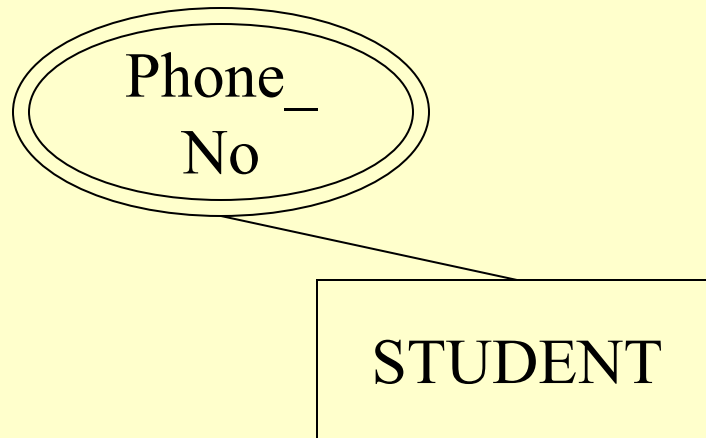
Types of Attributes

- **Atomic Attribute** is an attribute which can't be further subdivided into further meaningful attributes.
- For Ex: Price of a product, Gender.
- **Composite Attribute** is an attribute which can be broken into further meaningful attributes.
- For Ex: Address of a student.



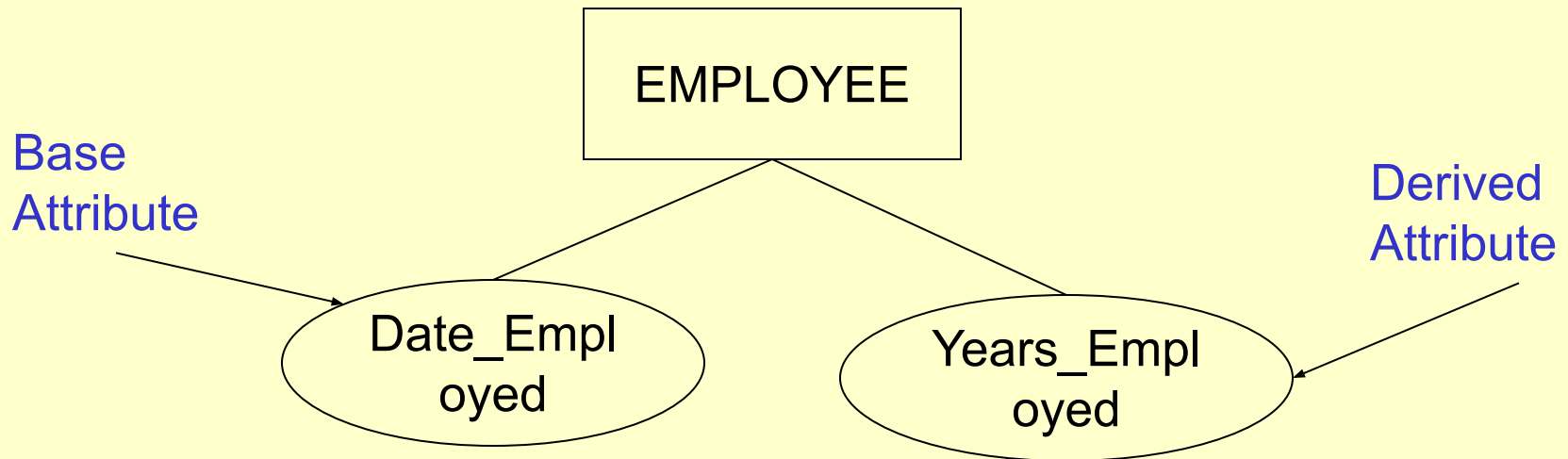
Types of Attributes

- **Single Valued Attribute** is an attribute which has one and only one value against an entity instance.
- For Ex: RegNo of a Student, Pno of a product etc.
- **Multi valued Attribute** is an attribute that may take on more than one values for a given entity instance.



Types of Attributes

- **Base Attribute** is an attribute that is stored in the database.
- **Derived Attribute:** An attribute whose value can be calculated from related attribute values.



Relationship

- A relationship describes an association among attributes of an entity and among different entities.
- The relationship between entities is represented using Data Structure Diagram (Batchman Diagram) and Entity Relationship Diagram (ERD).

Types of Relationship

- There are the following types of relationships or data associations:
 - a) One to One Association
 - b) One to Many Association
 - c) Many to Many Association

a) One to One Relationship (1:1, 1..1)

- Let A and B are two data items, then one to one association from A to B means that for a given value of A, there is one and only one value of B.
- Example: -----
- The special cases of one to one association are Bidirectional Association and Conditional Association.
- In a Bidirectional association, from a given value of any data item, we can uniquely determine the value of other.
- In Conditional Association, for a given value of A, there is either one or no value of B.

b) One to Many Relationship (1:M, 1..*)

- Let A and B are two data items, then one to many association from A to B means that for a given value of A, there can be many values of B.
- Example: -----

c) Many to Many Relationship (M:N, *..*)

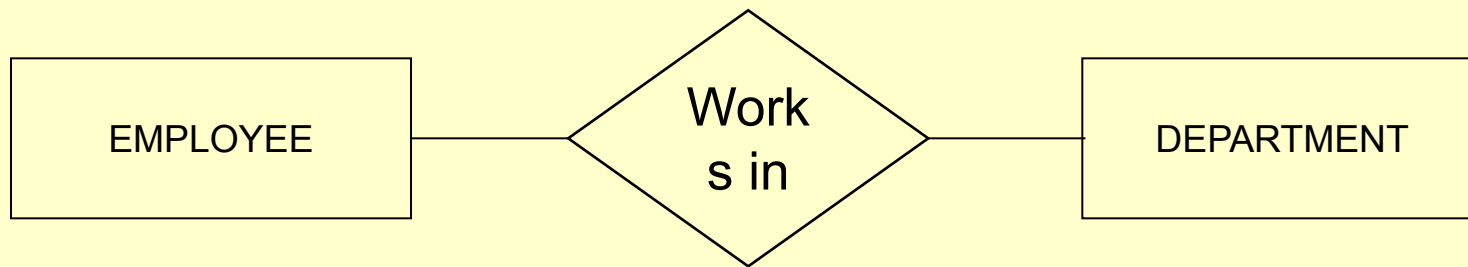
- Let A and B are **two** data items, then many to many association from A to B means that for a given value of A, there can be many values of B and vice-versa.
- Example: -----

Data Association Diagrams

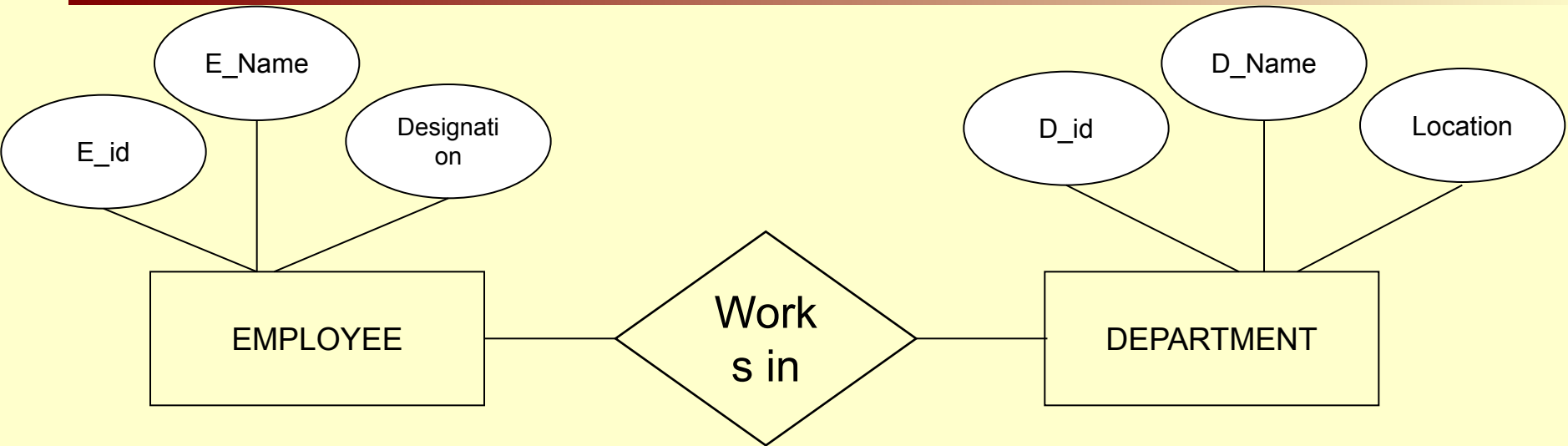
b) Data Structure Diagram or Batchman Diagram

- It is the graphical representation of the association/relationship between entities of an organization.
- This diagram is very helpful for conceptual understanding.
- Example: -----

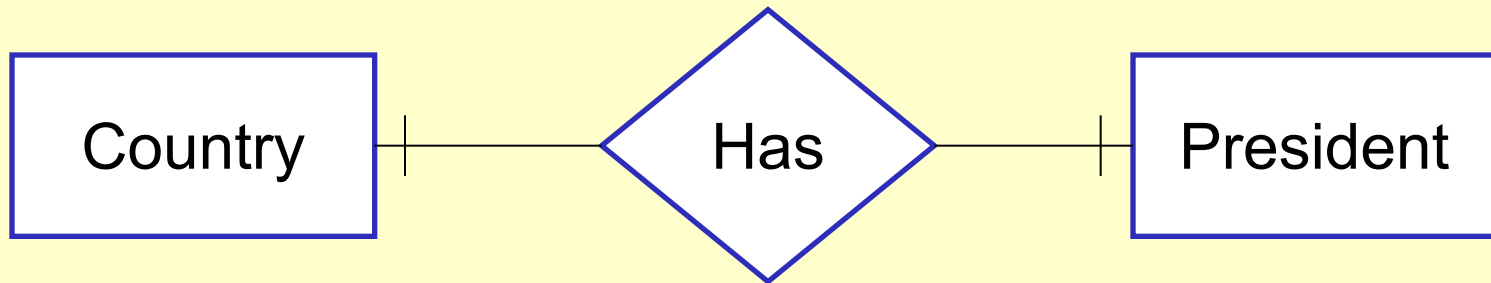
Entity Relationship (E-R) Diagram



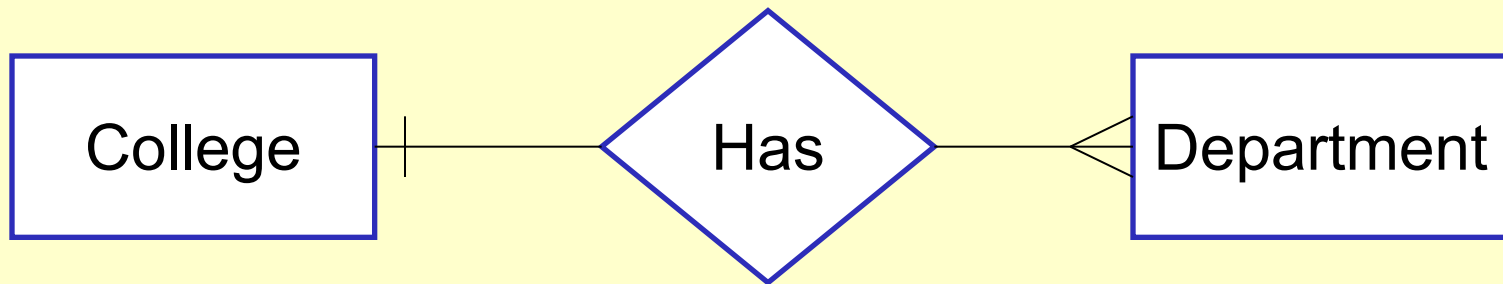
Entity Relationship (E-R) Diagram



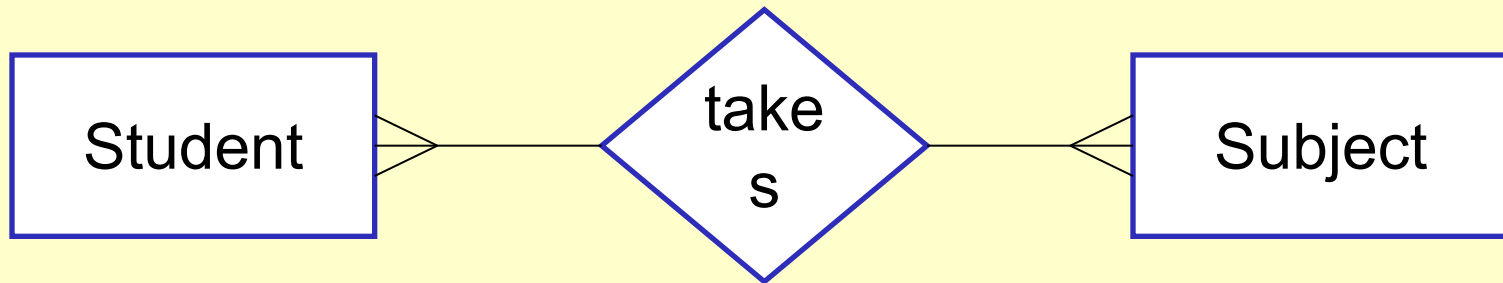
a) One to One Relationship (1:1, 1..1)



b) One to Many Relationship (1:M, 1..*)



c) Many to Many Relationship (M:N, *.*.)



Constraints

- A constraint is a restriction place on data.
- Constraints are used to improve data quality and data integrity.
- Examples ...