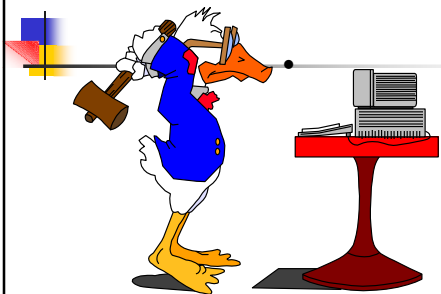


## The Entity-Relationship Model

### Conceptual Data Modeling



### The Entity-Relationship model

- The E-R model is a detailed, logical representation of the data for an organisation or business area
- It should be understandable to both the user and to the IT technologist

### The Entity-Relationship model

- The model must be as 'open' as possible and not tied to any technology or to any particular business methodology
- It must be flexible enough so that it can be used and understood in practically any environment where information is modelled

### The ER model

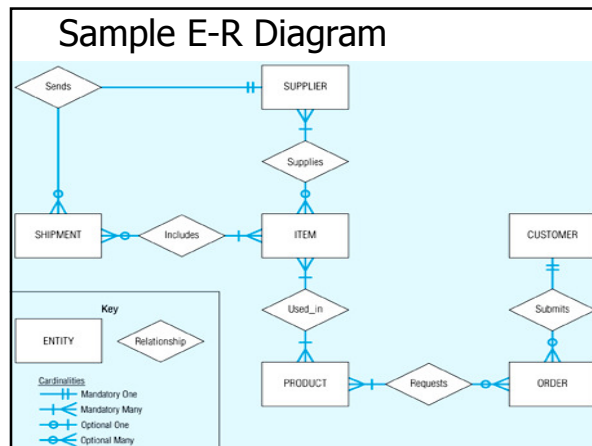
- It is expressed in terms of entities in the business environment, the relationships (or associations) among those entities and the attributes (properties) of both the entities and their relationships
- The E-R model is usually expressed as an E-R diagram

### E-R Model Constructs

- Entity - person, place, object, event, concept
- Each entity is given a name, since this name represents a set of items, it is always singular.
- It is placed inside the box representing the entity

### E-R Model Constructs

- Entity instance – is a single occurrence of an entity type.
- Eg Entity is EMPLOYEE and Instance is John Banda



## Strong versus Weak entity type

- A strong entity types – one that exists independently from other entity types
- A weak entity type – existence depends on some other entity type.
- It has no meaning in the ER diagram without the entity on which it depends
- The entity type on which the weak entity type depends is called the Identifying owner (or owner for short).

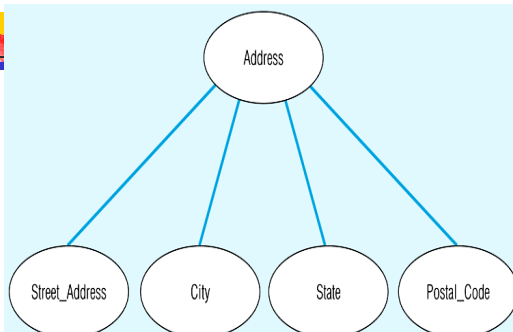
## Attributes

- An attribute is a property or characteristic of an entity, for example the entity EMPLOYEE may have attribute Employee\_Name
- Attributes may also be associated with relationships
- An attribute is associated with exactly one entity or relationship

## Simple versus composite attributes

- Some attributes can be broken down into meaningful component parts, such as Address, which can be broken down into Street Address, City..etc.
- A simple (atomic) attribute is one that cannot be broken down into smaller components

A composite attribute



## Single-Valued versus Multivalued Attribute

- It frequently happens that there is an attribute that may have more than one value for a given instance, e.g. EMPLOYEE may have more than one Skill.
- A Multivalued attribute is one that may take on more than one value.

## Derived Attributes

- Some attribute values can be calculated or derived from others
- e.g., Years\_Employed can be calculated for EMPLOYEE, by using Date\_Employed and Today's\_Date
- A derived attribute is one whose value can be calculated from related attribute values

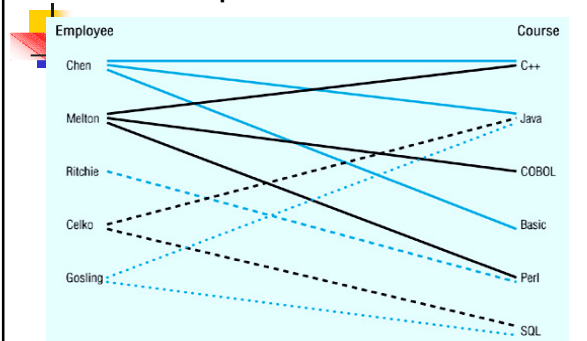
## Relationships

- A relationship is an association among the instances of one or more entity types that is of interest to the organisation
- Relationship Type is a meaningful association
  - implying that the relationship allows us to answer questions that could not be answered given only the entity types

## Relationship instance

- Is an association between (or among) entity instances, where each relationship includes exactly one entity from each participating entity type.
- For example, in the following figure each line represents a relationship instance between one employee and one course

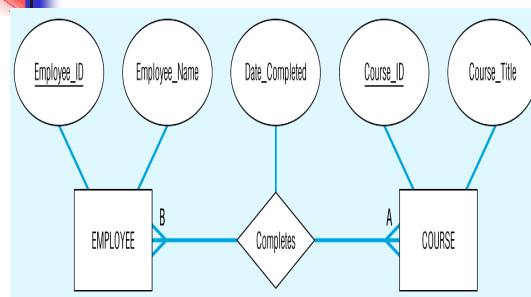
## Relationship instances



## Attributes on relationships

- Attributes may be associated with a many-to-many (or one-to-one) relationship, as well as with an entity
- In the following diagram, the relationship 'Completes' joins the EMPLOYEE and COURSE entities, and Date\_Completed is joined to this as it is a property of the relationship 'Completes'

## Attribute on a relationship



## Degree of a relationship

- number of entity types that participate in it.
- The three most common relationship degrees are unary (degree 1), binary (degree 2) and ternary (degree 3 –see following Fig.)
- Higher degree relationships are possible but rarely encountered in practice

## Unary relationship

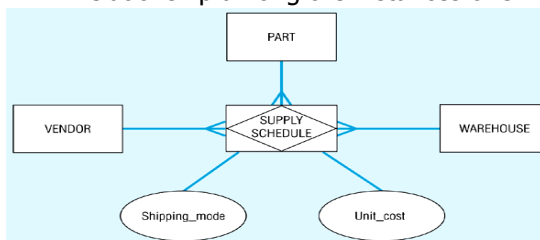
- Is between the instances of a single entity type (also called recursive relationships)
- 'Is\_Married\_To' is a relationship between instances of the PERSON entity type
- 'Manages' is a relationship between instances of the EMPLOYEE entity type

## Binary relationships

- Between the instances of two entity types, and is the most common type of relationship encountered in data modelling.

## Ternary relationships

- A ternary relationship is a simultaneous relationship among the instances of 3



## Cardinality constraints

- The number of instances of one entity that can or must be associated with each instance of another entity.
- e.g. a video store may stock more than one VIDEOTAPE for each MOVIE, this is a 'one-to-many' relationship as in the following.

## Minimum cardinality

- The minimum cardinality of a relationship is the minimum number of instances of an entity B that may be associated with each instance of an entity A
- In our example, the minimum number of VIDEOTAPES of a MOVIE.

### Maximum cardinality

- Is the maximum number of instances of an entity B that may be associated with each instance of entity A
- In the following Fig., the maximum cardinality for the VIDEOTAPE entity type is 'many' (an unspecified number greater than 1)

### Participation Constraints

- Is about the importance of the instances' *participation* in a specific relationship.
  - If it is applied to every instance of an entity, then it is called a *total* or a *mandatory participation*
  - (it is a *must* for each instance belongs to that entity type).

### Participation Constraints

- If only parts of the instances participate in a relationship or in other words, an instance may or may not participate in that relationship, then this is called a *partial* or *optional participation*.

### Example of participation Constraints

- EMPLOYEE Is\_Assigned\_To PROJECT
- Each PROJECT has at least one EMPLOYEE assigned to it.
- Each EMPLOYEE may or (optionally) may not be assigned to any existing PROJECT, or may be assigned to one or more PROJECTs.

