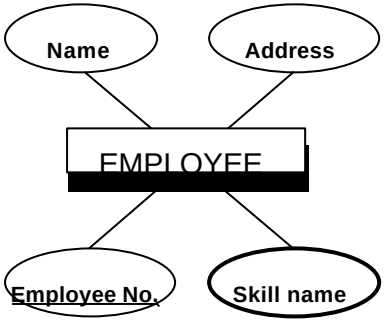
Database Modeling (Part 1)

Entity-Relationship Model

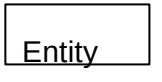
- A conceptual data model, which is a representation of the structure of a database that is independent of the software that will be used to implement the database

Entity-Relationship Model



Entity

- An Entity can be a person, place, object, event or concept in the user environment about which the organization wishes to maintain data.
- Examples:
  - Person: EMPLOYEE, STUDENT, PATIENT
  - Place: STATE, REGION, COUNTRY
  - Object: MACHINE, BUILDING
  - Event: SALE. REGISTRATION
  - Concept: ACCOUNT, COURSE



Entity Type and Entity Instance.

- There is a difference between **Entity Type** and **Entity Instance**.
  - Entity Type is a collection of entities that share *common* properties or characteristics
  - Entity Instance is a single *occurrence* of an entity type

Examples

- Entity type: EMPLOYEE
- Attributes:
  - EMPONO
  - NAME
  - ADDRESS
  - YEARHIRED
- Instances:

100	101
Roy Lim	Mary Wong
Hong Kong	Kowloon
1989	1999

Attribute

- Attribute is a property or characteristic of an entity that is of interest to the organization.
- Both entity and relationships may have attributes.
- Example:
  - STUDENT: STUDENT NO.,NAME, ADDRESS
  - EMPLOYEE: EMPLOYEE NO., NAME, SKILL

Candidate Key

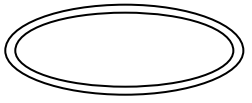
- Candidate key is an attribute (or combination of attributes) that uniquely identifies each instance of an entity type

Primary Key

- Primary Key is a candidate key that has been selected as the identifier for an entity type.

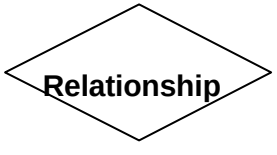
Multivalued Attribute

- Attribute that can have more than one value for each entity instance.
- For example
  - An EMPLOYEE may possess a number of skills.
  - So SKILL is a multivalued attribute.



Relationship

- Relationships are what that holds together the various entities..



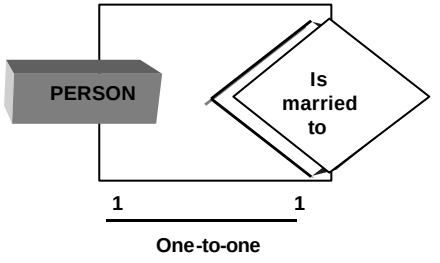
Degree of Relationship

- The degree of a relationship is the number of entity types that participate in that relationship
  - Unary relationship (recursive relationship)
  - Binary relationship
  - Ternary relationship

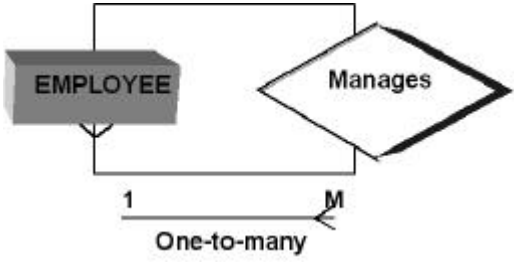
Unary Relationship

- Unary Relationship also called **Recursive Relationship**
- It is a relationship between the instances of one entity type

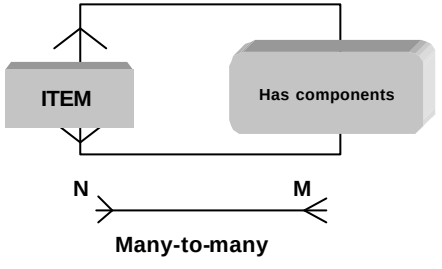
Unary Relationship (Recursive Relationship)



Unary Relationship (Recursive Relationship)

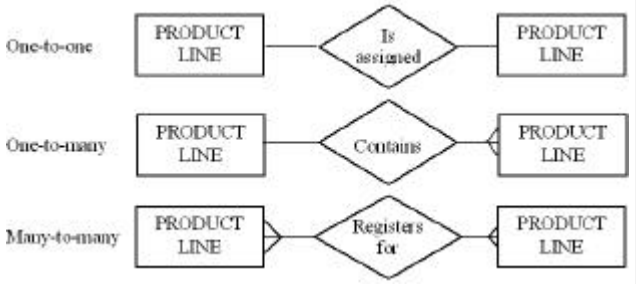


Unary Relationship (Recursive Relationship)



Binary Relationship

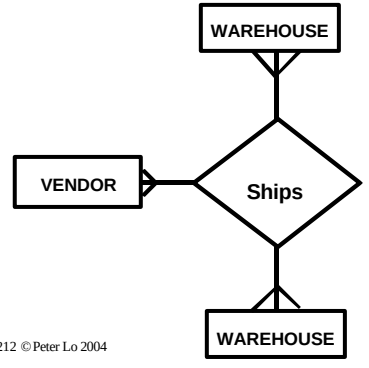
- It is a relationship between instances of two entity types.



Ternary Relationship

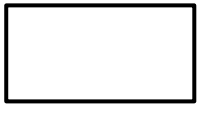
- It is a simultaneous relationship among instances of three entity types.
- Ternary relationship is a simultaneous relationship among instances of three entity types

Ternary Relationship



Existence Dependency

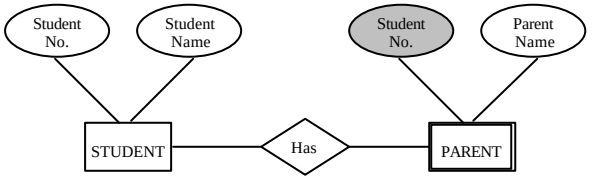
- An instance of one entity cannot exist without the existence of an instance of some other entity.



Weak Entity

- A Weak Entity is an entity type that has an existence dependency.
- A relationship in which the primary key of the parent entity is used as part of the primary key of the dependent entity.
- Weak entities usually do not have a natural identifier because the primary key of the parent entity is often used as part of the primary key of the dependent child entity.
- Data integrity is enforced as weak entity cannot exist unless the parent entity exists.

Example of Weak Entity



Generalization

- Generalization is the concept that some things (entities) are subtypes of other, more general things.
- To express generalization relationships, object are arranged into a hierarchy.
  - Example: Accountant, Programmer Analyst are subtypes of the more general type called STAFF.

Categorization

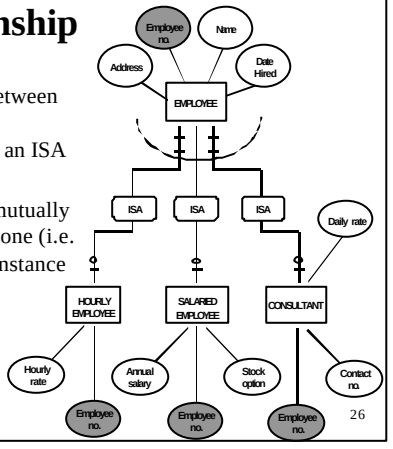
- Categorization is when an entity comes in various subtypes.
  - Example: There are different subtypes of employee which are hourly employee, salaried employee and part-time employee.

Supertypes and Subtypes

- Supertypes
  - A generic entity type that is subdivided into subtypes.
- Subtypes
  - A subset of a supertype that shares common attributes or relationships distinct from other subsets.

ISA Relationship

- The relationship between each subtype and supertype is called an ISA relationship.
- The subtypes are mutually exclusive and that one (i.e. required for each instance of the supertype).



Logical Database Design

- There is a process of transforming the conceptual data model into a logical database model.
- There are four types of logical database models in use today:
  - Object-oriented
  - Hierarchical
  - Network
  - Relational.

Object-Oriented Data Model

- Most future database management systems will be based on objects, or will incorporate object-oriented functionality.
- This enable users to create generic, all purpose components that can be reused in multiple applications.

Object

- Objects are abstraction of the real world entities that exhibit states and behaviors.
- The state of objects are expressed as values of the **Attributes** of the object.
- The behavior of an object is expressed by a set of **Methods** that operate on its attributes.

Attribute

- These are the properties of objects that are of interest to the organization.
- Attributes describe the characteristics of an object.
- Objects can have a specific attribute called a state.
- The state of an object describes the object's current status.

Methods

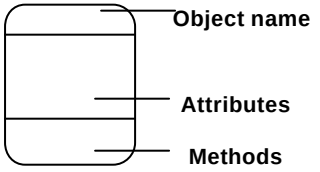
- Methods define the behavior of an object.
- Can only process data within the object class in which they are defined (Encapsulation).
- Can receive requests from methods in another object class.
- A number of method categories:
  - Occur Methods – instance add, instance change, instance delete.
  - Calculate Methods – perform calculations on the data values encapsulated in the same object class.
  - Monitor Methods – produce signals when predetermined limits are exceeded in a system.

Encapsulation

- Encapsulation is the property that the attributes and the methods of an object are hidden from the outside world and do not have to be known to access its data values or use its methods.

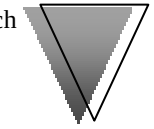
Object Classes and Instance

- A logical grouping of objects that have the same attributes and behavior.
- An object instance is one occurrence of an object class.
- When we use the term object by itself, we are referring to an object class.

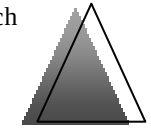


Identifying and Describing Objects

- Top-down Approach



- Bottom-up Approach



Top-down Approach

- This begins with a high level description of the environment and proceeds from the general to specific.
- In the object-oriented data model, methods are activated by sending message from a sending object to a receiving object.
- For example studying written material and talking with users are activities to locate nouns in written material.
- This provides Information about Potential Objects.

Bottom-up Approach

- Bottom-up approach begins with system detail, example reports, video forms and other detail documents and displays.
- The analyst identifies the Candidate Objects and their Properties.

Top-down vs. Bottom-up Approach

- In reality, the top-down and bottom-up approaches should be used to identify and describe candidate objects.

Inheritance

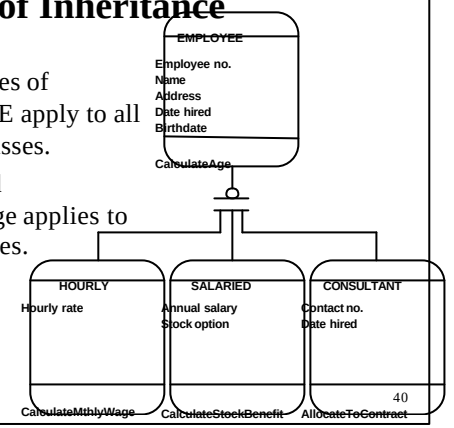
- Inheritance is an important principle of the object-oriented mode.
- Inheritance means that all properties of an object class become the properties of its subclasses.

Inheritance

- Inheritance is the property that, when entity types or object classes are arranged in a hierarchy, each entity type of object class assumes the attributes and methods of its ancestors.
  - All properties of an object class become the properties of its subclasses.
- Only attributes that are unique to a subtype are associated with that subtype.

Example of Inheritance

- The attributes of EMPLOYEE apply to all three subclasses.
- The method CalculateAge applies to all employees.



Advantage of OO Approach

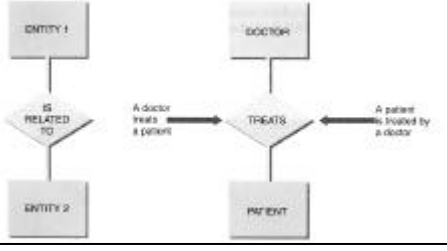
- Reusability
  - Objects can be defined for a variety of functions and then reused in numerous applications.
- Complex data types
  - An object-oriented database can store and manage complex data such as documents, graphics, images, voice message and video sequences.
- Distributed databases
  - Object-oriented databases can support distribution of data across a network more easily than other data models.

Appendix

Entity-Relationship Diagrams

Entity-Relationship Diagrams

- An Entity-Relationship Diagram (ERD) is a graphical model of the information system that depicts the relationships among system entities.



Three Main Types of Relationships

- One-to-One relationship (1:1) ⇨
- One-to-Many relationship (1:M) ⇨
- Many-to-Many relationship (M:N) ⇨

One-to-one relationship (1:1)

- It exists when exactly one of the second entity occurs for each instance of the first entity.

One-to-Many Relationship (1:M)

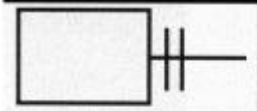
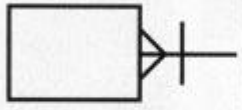
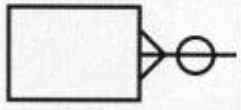
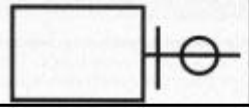
- It exists when one occurrence of the first entity can be related to many occurrences of the second entity, but each occurrence of the second entity can be associated with only one occurrence of the first entity.

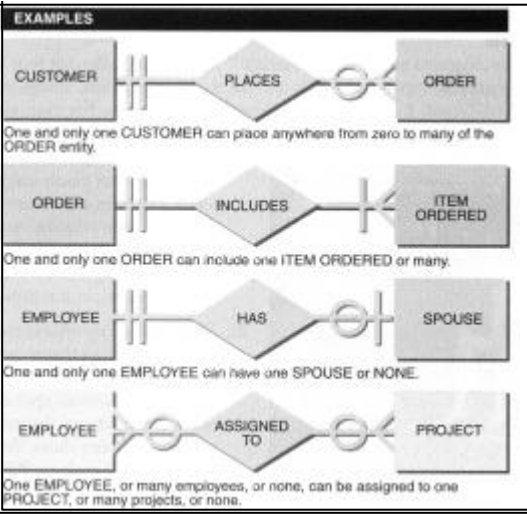
Many-to-Many Relationship (M:N)

- It exists when one instance of the first entity can be related to many instances of the second entity, and one instance of the second entity can be related to many instances of the first entity.

Cardinality

- Cardinality describes how instances of one entity relate to instances of another entity.

SYMBOL	MEANING	UML REPRESENTATION
	One and only one	1
	One or many	1..*
	Zero, or one, or many	0..*
	Zero, or one	0..1



Creating an Entity Relationship Diagram (ERD)

1. Identify the entities.
2. Determine all significant events, transactions, or activities that occur between two or more entities.
3. Analyze the nature of the interaction.
4. Draw the Entity Relationship Diagram.