**Modeling Notations**

A summary of the syntax of four common data modeling notations: Information Engineering (IE), Barker, IDEF1X, and the [Unified HYPERLINK "http://www.agilemodeling.com/essays/umlDiagrams.htm"Modeling HYPERLINK "http://www.agilemodeling.com/essays/umlDiagrams.htm" Language (UML)](http://www.agilemodeling.com/essays/umlDiagrams.htm)



Types of Entities

* Strong Entity
* Weak Entity
* Composite Entity
* Recursive Entity
* Super type and Sub type Entity

**Strong Entity :**

The Entity with key attributes is said to be strong entity.

**Weak Entity :**

The Entity without key attributes is said to be weak entity.

e.g. Employee (Strong) & Dependant (Weak)

Student (Strong) & Course (Weak)

**Composite Entity :**

The Composite Entity is used to convert many to many

relationship to one to many relationship

e.g Bridge table for Student Course mapping

**Recursive Entity :**

It is also called self referential relationship entity. It is an entity type foreign key referencing same table or itself. Emp Manager

**Super type Entity & Sub type Entity :**

A subtype is sub grouping of entities in an entity type that is meaningful to the organization.

Eg: Student Entity is Super type where as graduate student and under graduate student is sub type

* Let us assume an entity named EMPLOYEE. The entity employee has some attributes-empID, Name, Birth Date, Age etc.  
  
* But sometimes we will have special employees that have some different attributes to other employees.
* Imagine the employees at airline company, there are different type of employees including Accountants, Mechanics, Pilots, Flight attendants etc. A quick solution would be to include another attribute named “type” to the entity EMPLOYEE.  
  
* But the type attribute would work for some cases and in some cases, it doesn’t work as it create problems.  
  For example, an accountant has a CPA(Certified Public Accountant) that no other employee type has.  
  A pilot has a flying qualification and a pilot licence that no other employee has.  
  A mechanic has a qualification of aircraft engineering that no other employee has.
* So, this type of situation can be handled using the concept of sub-type and super type because we can use sub-types to give unique attributes to those types of employees.  
  
* Each sub-type(child) will automatically inherit all the attributes from the super type(parent) and each sub-type have some unique attributes that a parent and the other sub-type doesn’t have.
* So EMPLOYEE-super-type(parernt)  
  and PILOT, ACCOUNTANT, MECHANICS, FLIGHT\_ATTENDANTS-sub-types(child)



Attributes

|  |  |
| --- | --- |
| Entity | Attributes |
| employee | Empno,ename,salary,address |
| books | Book\_id,ISBN,Author,Price |
| student | Student\_id,course |

Types of Attributes

* Simple attribute
* Composite attribute
* Derived attribute
* Single value attribute
* Multi value attribute
* Stored attribute
* Coded Attribute

**Simple attribute :**

Simple attributes are atomic values, which cannot be divided further. For example, a student's phone number is an atomic value of 10 digits.

Example: Passport number ,PAN number ,Adhaar number

**Composite attribute :**

Composite attributes are made of more than one simple attribute.

Example: A student's complete name may have first\_name and last\_name.

Address of a person can have door number,street,city,state,zipcode.

**Derived attribute :**

Derived attributes are the attributes that do not exist in the physical database, but their values are derived from other attributes present in the database.

Example: Calculation of Age from the DOB

Calculation of Total salary from salary and commission.

**Single-value attribute :**

Single-value attributes contains only single value.

Example: Employee number , Passport number

**Multi-value attribute :**

Multi-value attributes may contain more than one values.

Example: Educational qualification B.E,B.L

Telephone Number 9092112345

9940409993

Languages known : Hindi,English

**Stored attribute :**

Stored attributes is stored in db not a derived from anything.

Example: Stored Attribute ==> DOB, DOJ

**Coded attribute :**

Coded attribute values are like Gender, Marital status which has a code to indicate

Coded Attribute ==> Gender(m,f)

Coded Attribute ==> Marital Status(1,2,3,4)







