**Entities**

Entities are specific objects or things in the mini-world that are represented in the database.

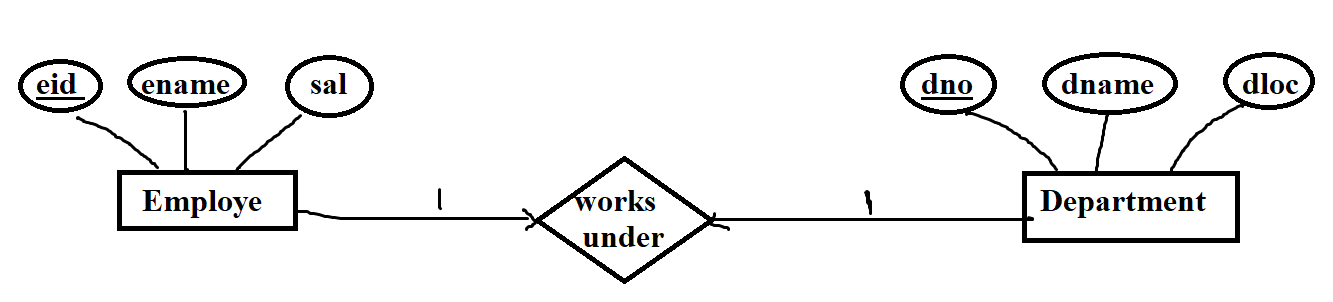
For example the STUDENT “Adam” (Adam is the entity instance & STUDENT is the entity type), the “Database” is the COURSE.

There are two types of entities:

* Strong entity
* Weak entity

**Strong Entity**

* A strong entity is not dependent on any other entity in the schema.
* A strong entity will always have a primary key.
* Strong entities are represented by a single rectangle.
* The relationship of two strong entities is represented by a single diamond.



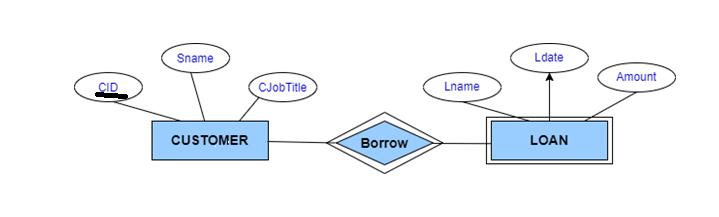
Example: EMPLOYE, DEPARTMENT each can be strong entity as each has its own primary key (eid (PK) For EMPLOYEE, dno for DEPARTMENT).

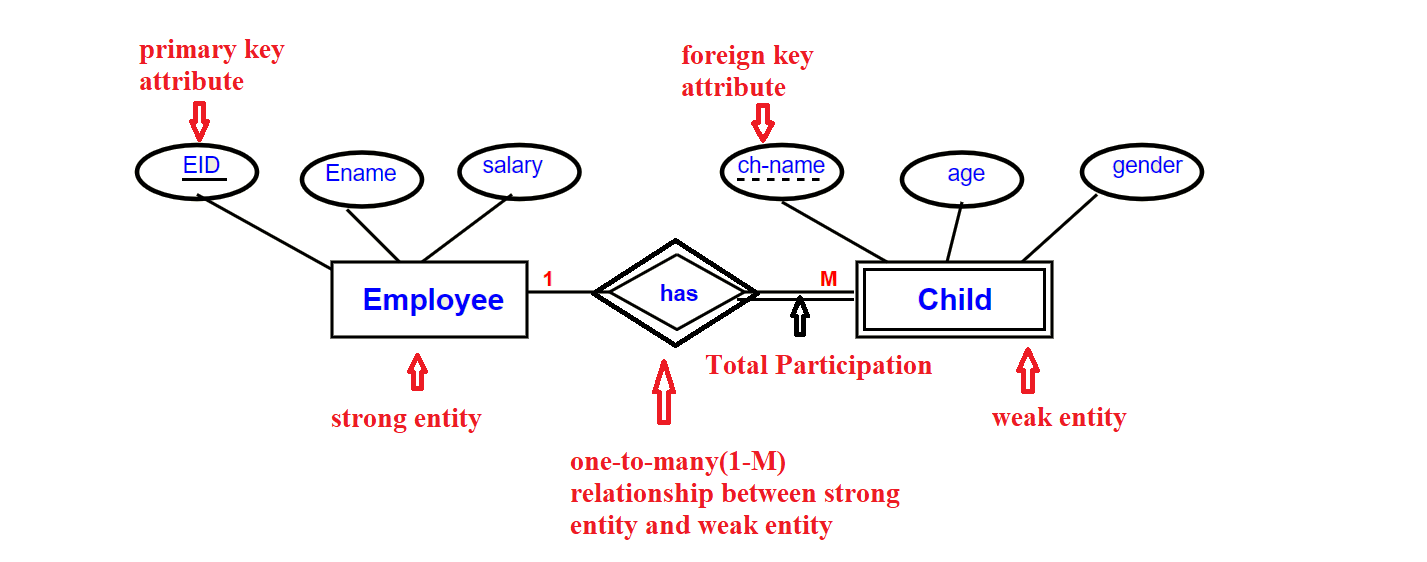
**Weak Entity**

* A weak entity is dependent on a strong entity to ensure its existence, Unlike a strong entity, a weak entity does not have any primary key.
* It instead has a partial discriminator key.
* A weak entity is represented by a double rectangle.
* The relation between one strong and one weak entity is represented by a double diamond.

Examples:

* CHILD entity depends on EMPLOYEE strong entity.
* LOAN entity depends on CUSTOMER strong entity.
* ROOM entity depends on HOTEL strong entity.





**Attributes**

* Attributes are properties of an entity, the attributes are used to describe an entity.
* For example an STUDENT entity may have a Name, ID, Address, Sex, Birthdate.
* A specific entity will have a value for each of its attributes.

**example:**

a specific “STUDENT” entity may have Name=’siva krishna’, ID=’123456789’, Address =’#210, annapurna block, Adithya enclave, ameerpet’, Sex=’M’, BirthDate=’14-jan-1991‘.

Each attribute has a value set (or data type) associated with it – e.g. integer, string, subrange, enumerated type, …,etc.,

**Types Of Attributes:**

There are numerous types of attributes, including the following:

* Simple Attribute & Composite Attribute
* Single Valued Attribute & Multi-valued Attribute
* Stored Attribute & Derived Attribute
* Key Attribute & Non-key Attribute

**Simple attribute:**

* A simple attribute has an atomic value that cannot be subdivided further.

Examples:

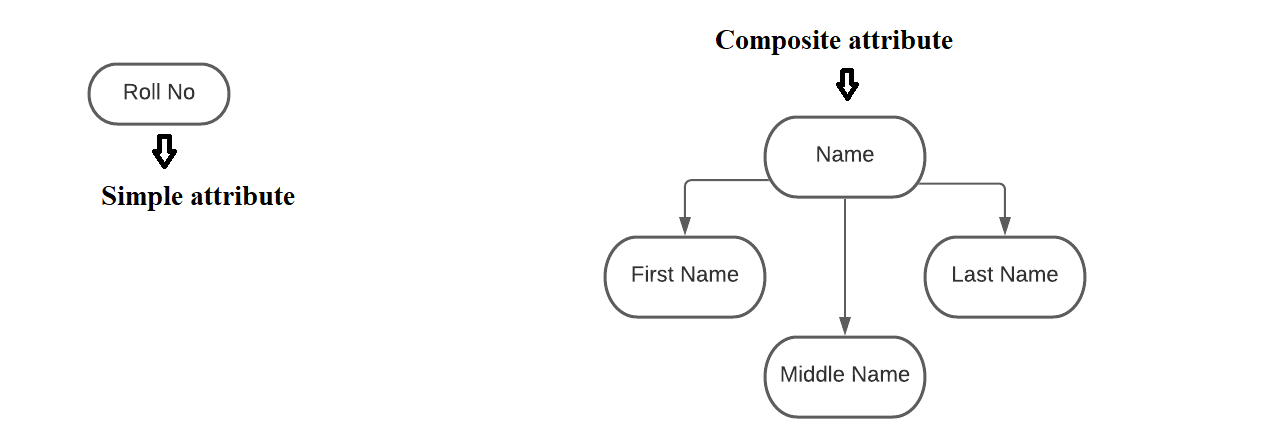
* The roll number of a student
* The age of a student
* The id number of an employee.

**Composite attribute:**

* A Composite attribute can be further divided into other attributes.

Examples:

* The address of a house can be further divided into the house number, street number, city, state, country, and pin code.
* The Name of a Student can be further divided into first name, middle name, and last name.



**Single valued Attribute:**

* A single-valued attribute is one that accepts only one value.

Example:

* a student cannot study in more than one section, it is a single-valued feature.

**Multi-valued Attribute:**

* A multi-valued attribute is one that can accept more than one value.

**Example:**

* A student can have multiple addresses, including both permanent and temporary addresses.

**Stored attribute:**

* The attribute is simply stored in the database.
* The user should give the value for this attribute.
* The value of this attribute is saved and can be used to find the value of other attributes.

Example:

* The date of birth of 'Student', this value must be provided by the student.

**Derived attribute:**

* This attribute's value is derived from the values of other characteristics.
* We know the value of another attribute (stored attribute) and are deriving the value of this attribute from the We stored attributes (derived attributes).

Example:

* The value of the age attribute is derived and extracted from the DOB(date of birth) attribute.

**Key attribute:**

* An entity type's entities are uniquely identified by a key attribute. A relational table represents the table's primary key.

Example:

* the "Roll No" of a student as a key attribute because a roll no cannot be the same for two students.

**Non-Key attribute:**

* The non-key attributes are all the other attributes that aren't the key attribute.
* This attribute can have the same value for two or more entities.

Example:

* all students enrolled in the same class would have the same value for the Class attribute.

