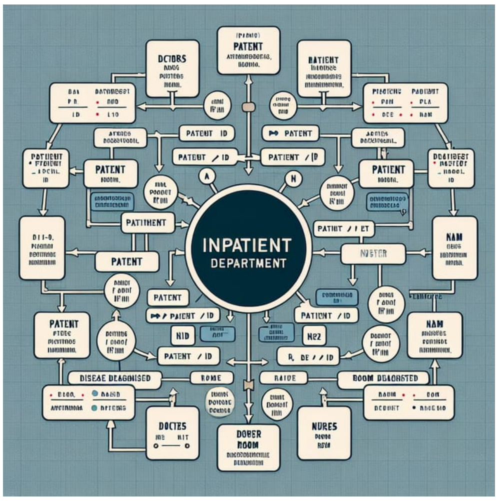
ENTITY RELATIONSHIP MODEL

Data Model

Collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints.



ER Model

It is a high-level data model based on a perception of a real world that consists of a collection of basic objects, called entities and of relationships among these objects.

Graphical representation of ER Model is ER diagram, which acts as a blueprint of DB.

Entity

An Entity is a “thing” or “object” in the real world that is distinguishable from all other objects. It has physical existence. Each student in a college is an entity.
Entity can be uniquely identified. (By a primary attributes, aka Primary Key)

Strong Entity: Can be uniquely identified.

Weak Entity: Can’t be uniquely identified, depends on some other strong entity.
It doesn’t have sufficient attributes, to select a uniquely identifiable attribute. Loan -> Strong Entity, Payment -> Weak, as instalments are sequential number counter can be generated separate for each loan. Weak entity depends on strong entity for existence.

Entity Set

It is a set of entities of the same type that share the same properties, or attributes.
E.g., Student is an entity set. Customer of a bank.

Attributes

An entity is represented by a set of attributes. Each entity has a value for each of its attributes. For each attributes, there is a set of permitted values, called the domain, or value set, of that attributes. E.g., Student Entity has following attributes: Student_ID, Name, Standard, Course, Batch, Contact number, Address.

Types of Attributes

- Simple Attributes which can’t be divided further. E.g., Customer’s account number in a bank, Student’s Roll number etc.

- Composite Attributes : Can be divided into subparts (that is, other attributes). E.g., Name of a person can be divided into first-name, middle-name, last-name. If user wants to refer to an entire attributes or to only a component of the attributes. Address can also be divided, street, city, state, PIN code.

- Single-Valued Attributes: Only one value attributes. E.g., Student ID, loan-number for a loan.

- Multi-Valued Attributes: Having more than one value. E.g., phone-number, nominee-name on some insurance, dependent-name etc. Limit constraint may be applied, upper or lower limits.
- Derived Attributes: Value of this type of attributes can be derived from the value of other related attributes. E.g., Age, loan-age, membership-period etc.

An attributes takes a null value when an entity does not have a value for it. It may indicate “not applicable”, value doesn’t exist. e.g., person having no middle-name. It may indicate “unknown”. Unknown can indicate missing entry, e.g., name value of a customer is NULL, means it is missing as name must have some value. Not known, salary attributes value of an employee is null, means it is not known yet.

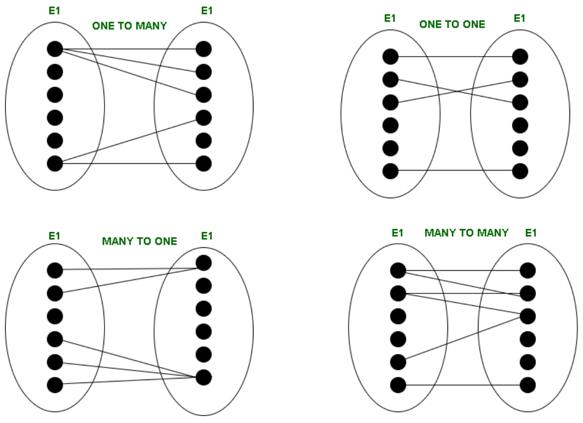
Relationships

Association among two or more entities. E.g., Person has vehicle, Parent has Child, Customer borrow loan etc. Strong Relationship, between two independent entities. Weak Relationship, between weak entity and its owner/strong entity. E.g., Loan <instalment-payments> Payment.

Degree of Relationship: Number of entities participating in a relationship.

- Unary, Only one entity participates. e.g., Employee manages employee.
- Binary, two entities participate. e.g., Student takes Course. Binaries are common.
- Ternary relationship, three entities participate. E.g. Employee works-on branch, employee works-on job.

Mapping Cardinality / Cardinality Ratio Number of entities to which another entity can be associated via a relationship.



- One to one, Entity in A associates with at most one entity in B, where A G B are entity sets. And an entity of B is associated with at most one entity of A. E.g., Citizen has Aadhar Card.

- One to many, Entity in A associated with N entity in B. While entity in B is associated with at most one entity in A. E.g. Citizen has Vehicle.

- Many to one, Entity in A associated with at most one entity in B. While entity in B can be associated with N entity in A. E.g. Course taken by Professor.

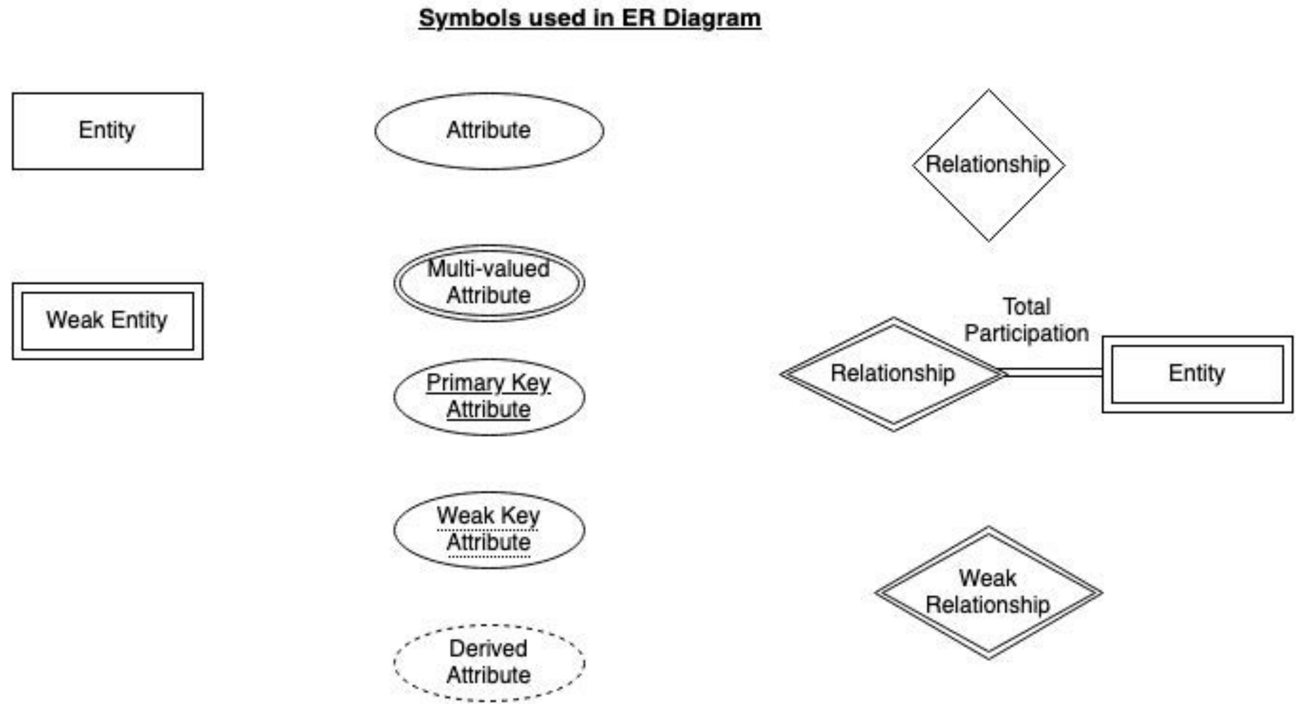
- Many to many, Entity in A associated with N entity in B. While entity in B also associated with N entity in A. Customer buys product. Student attends course.

Participation Constraints aka, Minimum Cardinality Constraint.

- Partial Participation, not all entities are involved in the relationship instance.
- Total Participation, each entity must be involved in at least one relationship instance. E.g. Customers borrow loan, loan has total participation as it can’t exist without customer entity. And customer has partial participation.

Note: Weak entity has total participation constraint, but strong may not have total.

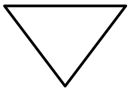
ER Notations



Specialization

In ER model, we may require to subgroup an entity set into other entity sets that are distinct in some way with other entity sets. Specialization is splitting up the entity set into further sub entity sets on the basis of their functionalities, specialties and features.
It is a Top-Down approach. E.g., Person entity set can be divided into customer, student, employee. Person is superclass and other specialized entity sets are subclasses.

We have **“is-a”** relationship between superclass and subclass. Depicted by triangle component.



Why Specialization?
Certain attributes may only be applicable to a few entities of the parent entity set. DB designer can show the distinctive features of the sub entities. To group such entities we apply Specialization, to overall refine the DB blueprint.

Generalization

It is just a reverse of Specialization. DB Designer may encounter certain properties of two entities are overlapping. Designer may consider making a new generalized entity set. That generalized entity set will be a super class.

“is-a” relationship is present between subclass and super class. E.g., Car, Jeep and Bus all have some common attributes, to avoid data repetition for the common attributes. DB designer may consider Generalizing to a new entity set “Vehicle”. It is a Bottom-up approach.