

DBMS NOTES

UNIT 2 - ER & EER Model

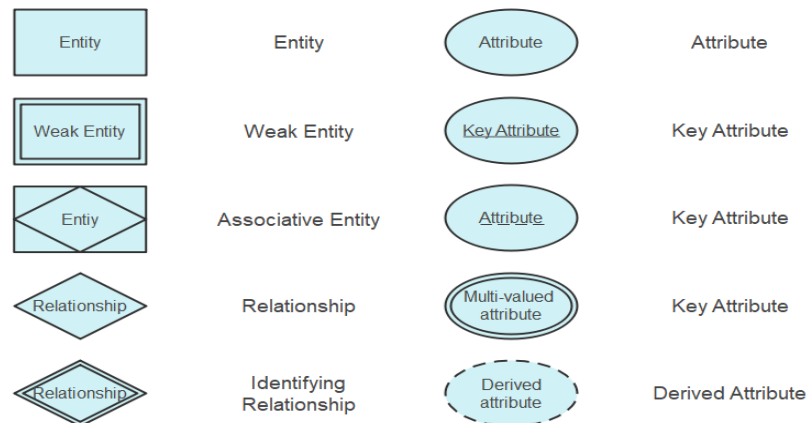
- **Data Model**

- It gives us an idea of how the final system will look after its complete implementation. It defines the data elements and the relationships between the data elements.
- Data Models are used to show how data is stored, connected, accessed and updated in the database management system.
- Here, we use a set of symbols and text to represent the information so that members of the organisation can communicate and understand it.
- Though there are many data models being used nowadays, the Relational model is the most widely used model.
- Some of the Data Models in DBMS are:
 - Hierarchical Model
 - Network Model
 - Entity-Relationship Model
 - Relational Model
 - Object-Oriented Data Model
 - Object-Relational Data Model
 - Flat Data Model
 - Semi-Structured Data Model
 - Associative Data Model
 - Context Data Model

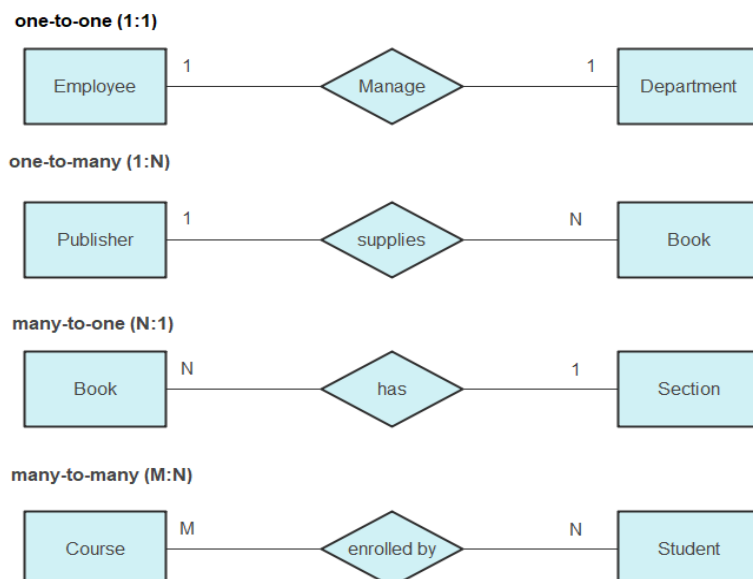
- **ER Model & its Basics -**

- ER Model is used to model the logical view of the system from data perspective which consists of these components - Entity, Entity Type, Entity Set, etc.
- Entity - It is a collection of objects. An entity is an object that is distinguishable from other objects by a set of attributes. This is the basic object of the E-R Model, which is a 'thing' in the real world with an independent existence. An entity may be an 'object' with a physical existence. Types of entity -
 - Strong Entity - Have a primary key for unique identification of rows.
 - Weak Entity - Entity types that do not have key attributes of their own are called as weak entity types. A weak entity type always has a 'total participation constraint'. A weak entity set can be identified uniquely only by considering some of its attributes in conjunction with the primary key of another entity (Identifying owner) and such relation called 'Identifying Relationship Set'.
- Entity Set - Set of entities of the same type that share the same property.
- Value Set or Domain - Available set of values for attribute.
- Attributes - Characteristics of an entity are called as an attribute. The properties of a particular entity are called as attributes of that specified entity. These can be classified into following types -
 - Simple Attributes - The attributes that are not divisible, example - grade, account_number.
 - Composite Attributes - Attributes that are not divisible, example - name into first & last name.
 - Single Valued Attributes - Attributes with a single value for a particular entity, example - Age
 - Multivalued Attributes - The attributes, which have a set of values for the same entity, example - College Degree.
 - Stored Attributes - An attribute which is not derived from another attribute.

- Derived Attributes - An attribute which is derived from another attribute.
- Descriptive Attributes - It is an attribute that describes or gives information about the relationship.



- Relationship - It describes the association among entities.
 - Types based on number of entity associated -
 - Unary / Recursive Relationship - Where there is only an entity which is related to itself.
 - Binary Relationship - Where there are 2 entities associated with some relationship with each other.
 - N-ary Relationship - Where there are 'n' different entities associated with some relationship with each other.
 - Types based on cardinality (It specified the maximum number of relationship instances) -
 - One to One (1:1) - When each entity in each entity set can take part only once in the relationship, the cardinality is one to one.
 - One to Many (1:M) - When entities in one entity set can take part only once in the relationship set and entities in another entity set can take part more than once in the relationship set, cardinality is many to one.
 - Many to Many (M:N) - When entities in all entity sets can take part more than once in the relationship cardinality is many to many.
- Relationship Sets - It is a collection of relationships.

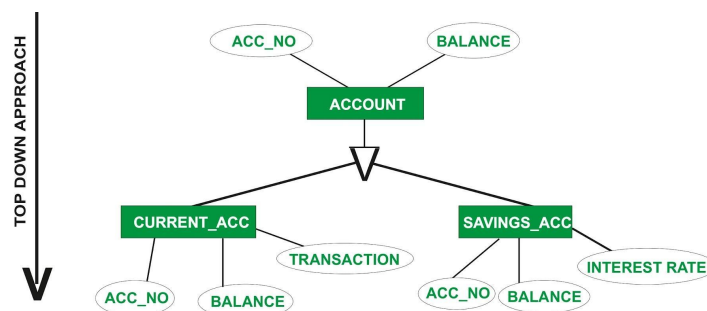


- Constraints - Some kind of restrictions placed on the data.
 - Participation Constraints - The participation constraint specifies whether the existence of an entity depends on its being related to another entity via the relationship type. Types -
 - Total - An entity set dependent on a relationship set and having one to many relationships is said to be 'total'. The participation of the entity set 'departments' in the relationship set 'manages' is said to be total.
 - Partial / Optional - A participation that is not total is said to be partial. Example - Doctor treats Patients. There can be a doctor with no patients.



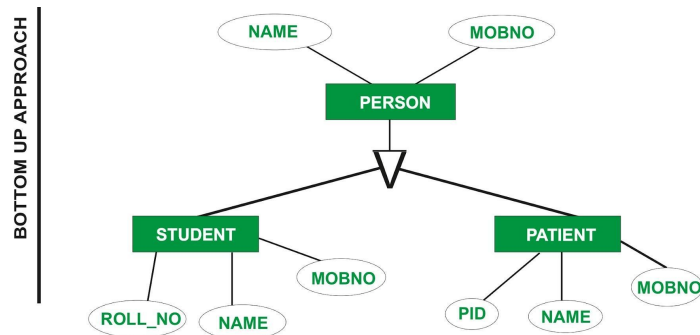
● EER Model & its Basics-

- Enhanced entity-relationship diagrams are advanced database diagrams very similar to regular ER diagrams which represent requirements and complexities of complex databases.
- Specialisation - It is a process of defining the set of sub classes of entity type or entity set. In specialisation things are broken down into smaller things to simplify it further. We can also say that in specialisation a particular entity gets divided into sub entities and it's done on the basis of its characteristics.
 - Also in specialisation Inheritance takes place.
 - Predicate Define - Those sub classes that can be determined by placing a condition on the values of certain attributes in a superclass. This condition is called Defining predicate. Example - Student with cgpa > 9 and cgpa <= 9.
 - Attribute Define - When membership conditions are applied on some attribute the subclasses form are called attribute define subclasses and the attribute is called defining attributes.



- Constraints in specialisation -
 - Disjoint Constraints - This ensures that there must be any entity present in more than one sub class hence, can be achieved through attributes defining specialisation.
 - Overlapping Constraints - When this is imposed there can be a common entity present in multiple subclasses. When specialisation type is predicate defined this is achieved.
 - Completeness Constraints -
 - Total specialisation
 - Partial specialisation

- Generalisation - It works on the principle of bottom up approach. In Generalisation lower level functions are combined to form higher level functions which are called entities. This process is repeated further to make advanced level entities.



- Aggregation - An ER diagram is not capable of representing the relationship between an entity and a relationship which may be required in some scenarios. In those cases, a relationship with its corresponding entities is aggregated into a higher level entity. Aggregation is an abstraction through which we can represent relationships as higher level entity sets.

