Entity Relationship Data Model
Unit – II

Entity Relationship Data Model was introduced in a key article by Chen (1976) in which he describe the main construct of the ER-Model. – Entities & relationships and their associates attribute. An Entity Relationship Model is a detailed logical representation of the Data for an organization or a business area. An Entity Relationship Model is normally expressed as an Entity Relationship Diagram.

Components of ER-Model:

1) **Entity –** An Entity is a person, place, object, event or concept in the real world i.e. distinguishable from all other objects.
2) **Entity Sets –** An Entity set of Entities of the same type that share the same properties or attributes.
    a. **Strong Entities –** A strong entity set is one that exists independent of other entity sets. A strong entity set that has primary key.
    b. **Weak Entities –** A weak entity is an entity whose existence depends on some other entities. A strong entity set that has no primary key.
3) **Attributes –** An entity can be simply defined as property or characteristics of an entity.
    a. **Simple Attribute –** Simple attributes is an attributes that cannot be broken into smaller subparts.
    b. **Composite Attribute –** Composite Attribute is an attributes that can be broken into smaller subparts.
    c. **Single Valued Attribute –** An attribute is said to be single valued attribute if it can have only one value.
    d. **Multi Value Attribute -** An attribute is said to be single valued attribute if it can have only more than one value.
    e. **Stored Attribute –** An attribute which is already present as an attribute for an entity is a stored attribute.
    f. **Derived Attribute -** An attribute which is derived from stored attribute as it is not present as an attribute for an entity is a derived attribute.
    g. **Null Attribute –** An attribute that can have null value is a null attribute.

Relation (or Table):

The terms Relation & Table can be used interchangeably. Each relation consists of a set of named columns. An attribute is a named column of a relation.

A relation has the following properties:-

1) In any given column of a table, all items are of the same kind whereas items in different columns may not be of the same kind.
2) For a row, each column must have an atomic (indivisible) value and also for a row, a column cannot have more than one value.
3) All rows of a relation are distinct. That is, a relation does not contain two rows which are identical in every column. That is, each row of the relation can be uniquely identified by its contents.
4) The ordering of rows within a relation is immaterial. That is, we cannot retrieve any things by saying that from row number 5, column name is to be accessed. There is no order maintained for rows inside a relation.
5) The columns of a relation are assigned distinct names and the ordering of these columns is immaterial.

Employee

EmpID	Name	DeptName	Salary
1001	Ravindra Agrawal	Finance	20000
1002	Khelan Nagar	Production	18000
1003	Himanshu Kulkarni	Personnel	25000
1004	Amol Maheshwari	Marketing	30000
1005	Ritesh Singh Chouhan	Advertisement	22000

Relationship Sets:

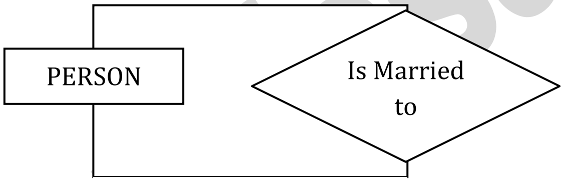
A relationship is an association among several entities. Relationships are the glue that holds together the various components of an ER Model.

A relationship set is a set of relationships of the same type. For example in a bank, any customer can have any types of loan (Business loan, Personal loan, Home loan) given by the bank. So all the relationship between all the customers and the loan taken by them are together called as relationship set.

Degree of Relationship:

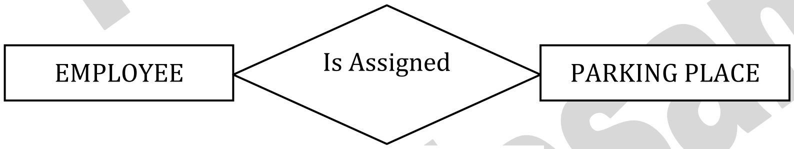
The degree of a relationship is the number of entity types that participate in that relationship. The three most common relationships in E-R-Model are Unary (degree 1), Binary (degree 2) and Ternary (degree 3).

1) **Unary Relationship**: A unary relationship is a relationship between the instance of a single entity type.

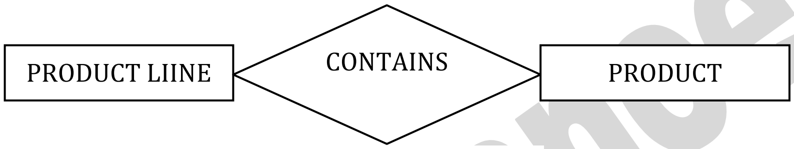


ONE-TO-ONE

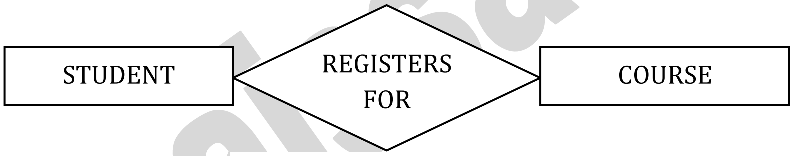
2) **Binary Relationship**: A binary relationship is a relationship between the instances of two entity types and is the most common type of relationship encountered in data modeling.



ONE-TO-ONE

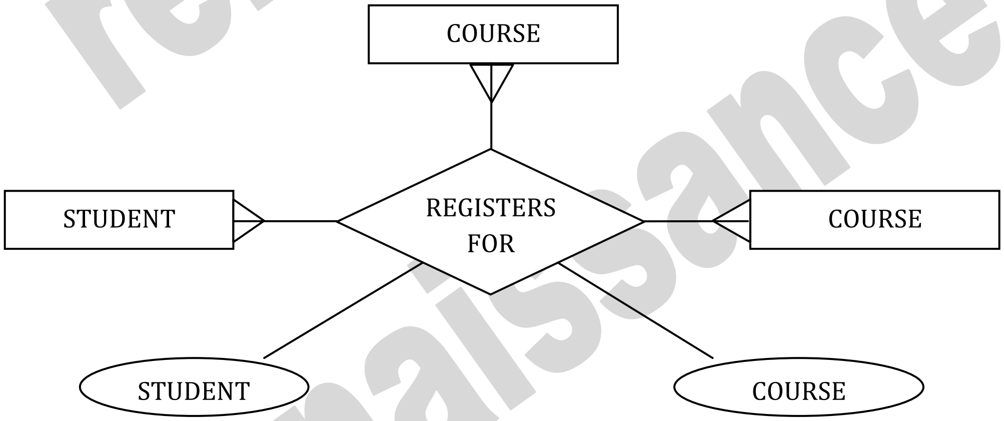


ONE-TO-MANY



MANY-TO-MANY

3) **Ternay Relationship**: A ternary relationship is a simultaneous relationship among the instances of three entity types.



Keys:

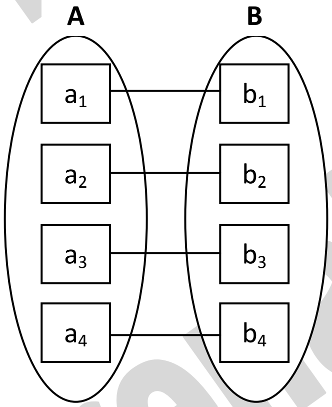
Keys are attributes or set of attributes used to distinguish one entity from another in an entity set.

1) **Super Key:** A super key is set of one or more attributes that can uniquely identify an entity in an entity set.
2) **Candidate Key:** All the attributes or set of attribute, when can uniquely identify an entity are candidate keys. Only those key can be candidate key whose no proper subset is a superkey.
3) **Primary Key:** The primary key is the term used for the candidates key that is chosen by the database designer as the principal means of identifying an entity.
4) **Alternate Keys:** The alternate key is term used for the candidate keys that are remaining after the primary key has be choosen by database designer.
5) **Foreign Key:** A foreign key is an attribute or set of attribute in a relation of database that serve as the primary key of another relation in the same database.
6) **Composite Key:** A primary key that consists of more than one attribute is called composite key.

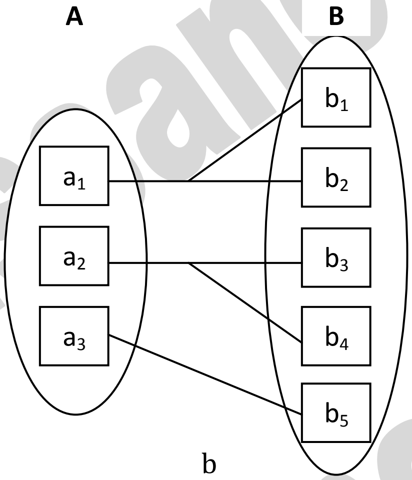
Mapping Constraints:
An E-R enterprise schema may define certain constraints to which the contents of a database must conform. This process can be termed as Mapping Constraints.
(1) **Cardinality Constraint**: Cardinality Constraint specifies the number of instances of one entity that can or must be associated with each instance of another entity.

Mapping cardinalities are most useful in describing binary relationship sets, although they can contribute to the description of relationship sets that involve more than two entity sets.

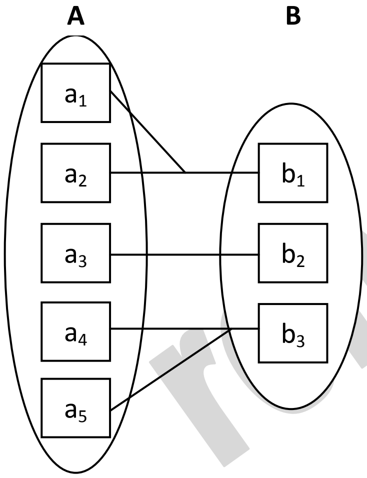
- **One to One:** An entity in A is associated with at most one entity in B and an entity in B is associated with at most one entity in A.
- **One to Many:** An entity in A is associated with any number (zero or more) of entities in B. An entity in B, however can be associated with at most one entity in A.
- **Many to One:** An entity in A is associated with at most one entity in B. An entity in B, however can be associated with any number (zero or more) of entities in A.
- **Many to Many:** An entity in A is associated with any number (zero or more) of entities in B and an entity in B is associated with any number (zero or more) of entities in A.

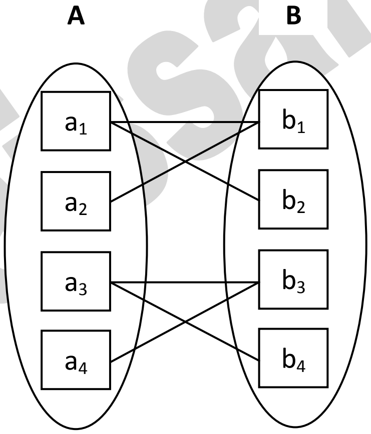


a



b





c d

(2) **Existence Dependencies**: Another important class of constraints is existence dependencies. Specifically, if the existence of entity x depends on the existence of entity y, then x is said to be existence dependent on y. If y is deleted so x has to be deleted. Entity y is said to be **dominant entity** and entity x is said to be **subordinate entity**.

Extended E-R Features:-

Although the basic E-R concepts can model most database features, some aspects of a database may be more aptly expressed by certain extensions to the basic E-R model. The extended E-R features are as follows: -

- specialization,
- generalization,
- higher- and lower-level entity sets,
- attribute inheritance, and
- Aggregation.

Specialization

An entity set may include subgroupings of entities that are distinct in some way from other entities in the set. For instance, a subset of entities within an entity set may have attributes that are not shared by all the entities in the entity set. The E-R model provides a means for representing these distinctive entity groupings.

Consider an entity set *person*, with attributes *name*, *street*, and *city*. A person may be further classified as one of the following:

· *customer*

· *employee*

Each of these person types is described by a set of attributes that includes all the attributes of entity set *person* plus possibly additional attributes. For example, *customer* entities may be described further by the attribute *customer-id*, whereas *employee* entities may be described further by the attributes *employee-id* and *salary*. The process of designating subgroupings within an entity set is called **specialization**. The specialization of *person* allows us to distinguish among persons according to whether they are employees or customers.

Generalization

The refinement from an initial entity set into successive levels of entity subgroupings represents a top-down design process in which distinctions are made explicit. The design process may also proceed in a bottom-up manner, in which multiple entity sets are synthesized into a higher-level entity set on the basis of common features. The database designer may have first identified a customer entity set with the attributes name, street, city, and customer-id, and an employee entity set with the attributes name, street, city, employee-id, and salary.

There are similarities between the customer entity set and the employee entity set in the sense that they have several attributes in common. This commonality can be expressed by generalization, which is a containment relationship that exists between a higher-level entity set and one or more lower-level

entity sets. In our example, person is the higher-level entity set and customer and employee are lower-level entity sets.
Higher- and lower-level entity sets also may be designated by the terms superclass and subclass, respectively. The person entity set is the superclass of the customer and employee subclasses.

Attribute Inheritance

A crucial property of the higher- and lower-level entities created by specialization and generalization is attribute inheritance. The attributes of the higher-level entity sets are said to be inherited by the lower-level entity sets. For example, customer and employee inherit the attributes of person. Thus, customer is described by its name, street, and city attributes, and additionally a customer-id attribute; employee is described by its name, street, and city attributes, and additionally employee-id and salary attributes. A lower-level entity set (or subclass) also inherits participation in the relationship sets in which its higher-level entity (or superclass) participates. The officer, teller, and secretary entity sets can participate in the works-for relationship set, since the superclass employee participates in the works-for relationship. Attribute inheritance applies
through all tiers of lower-level entity sets. The above entity sets can participate in any relationships in which the person entity set participates.
Whether a given portion of an E-R model was arrived at by specialization or generalization,
the outcome is basically the same:

- A higher-level entity set with attributes and relationships that apply to all of its lower-level entity sets
- Lower-level entity sets with distinctive features that apply only within a particular lower-level entity set

Aggregation

One limitation of the E-R model is that it cannot express relationships among relationships. To illustrate the need for such a construct, consider the ternary relationship works-on, which we saw earlier, between a employee, branch, and job Now, suppose we want to record managers for tasks performed by an employee at a branch; that is, we want to record managers for (employee, branch, job) combinations. Let us assume that there is an entity set manager. One alternative for representing this relationship is to create a quaternary relationship manages between employee, branch, job, and manager. (A quaternary relationship is required—a binary relationship between manager and employee would not permit us to represent which (branch, job) combinations of an employee are managed by which manager.)
It appears that the relationship sets works-on and manages can be combined into one single relationship set. Nevertheless, we should not combine them into a single relationship, since some employee, branch, job combinations many not have amanager. There is redundant information in the resultant figure, however, since every employee, branch, job combination in manages is also in works-on. If the manager were a value rather than an manager entity,we could instead make manager amultivalued attribute of the relationship works-on. But doing so makes it more difficult (logically as well as in execution cost) to find, for example, employee-branch-job triples for which
a manager is responsible. Since the manager is a manager entity, this alternative is ruled out in any case.
The best way to model a situation such as the one just described is to use aggregation. Aggregation is an abstraction through which relationships are treated as higherlevel entities.

E-R Diagram:

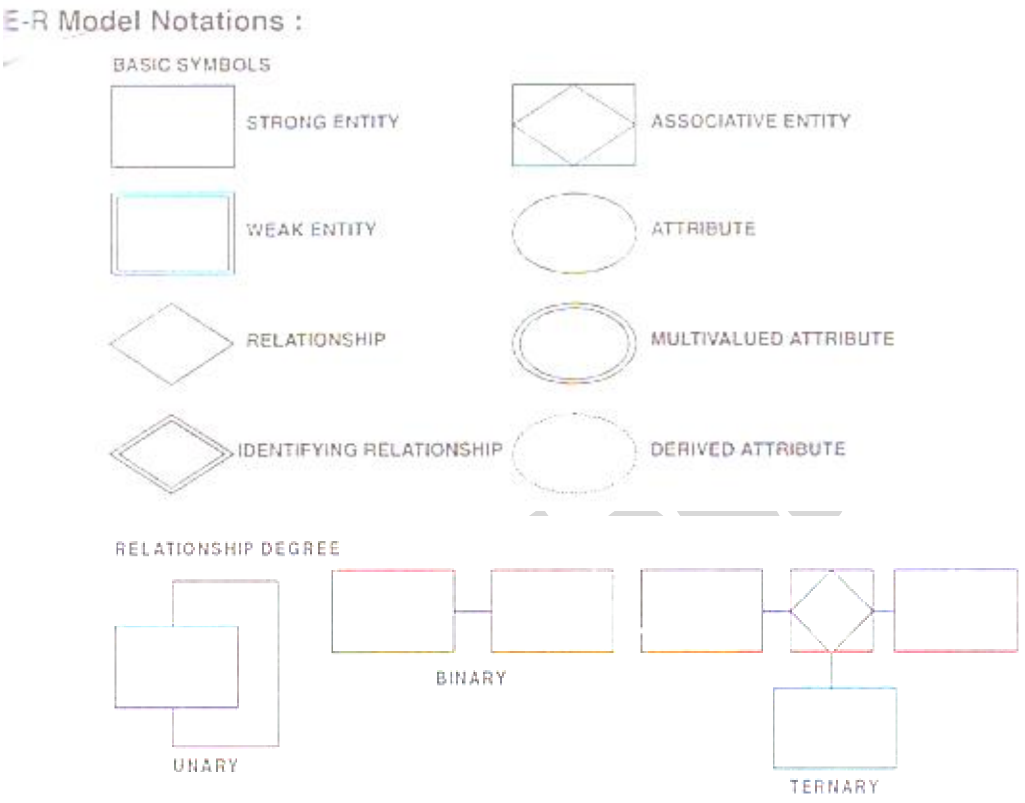
The basic E-R model first introduced during mid 1970s. It has been suitable for modeling most common business problems and has enjoyed widespread use.

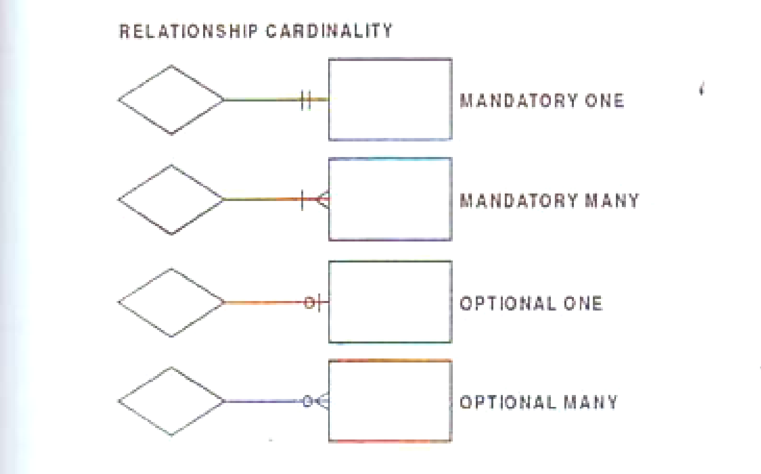
The overall logical structure of a database can be expressed graphically by an E-R diagram.

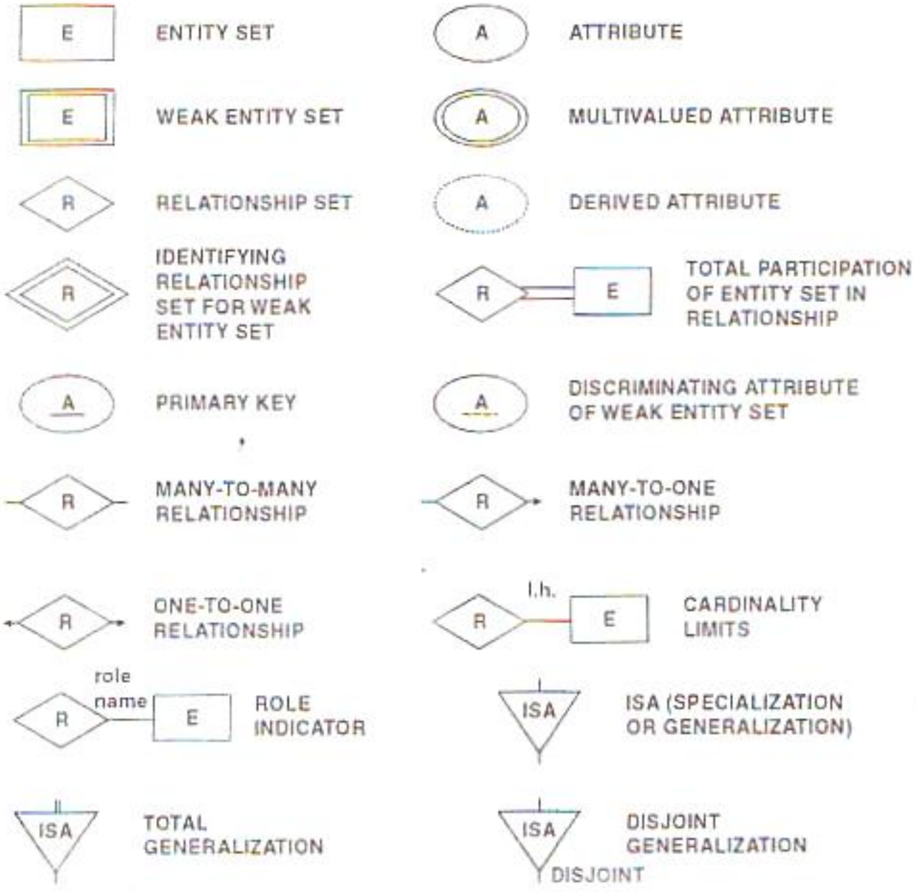
E-R Diagram Conventions:

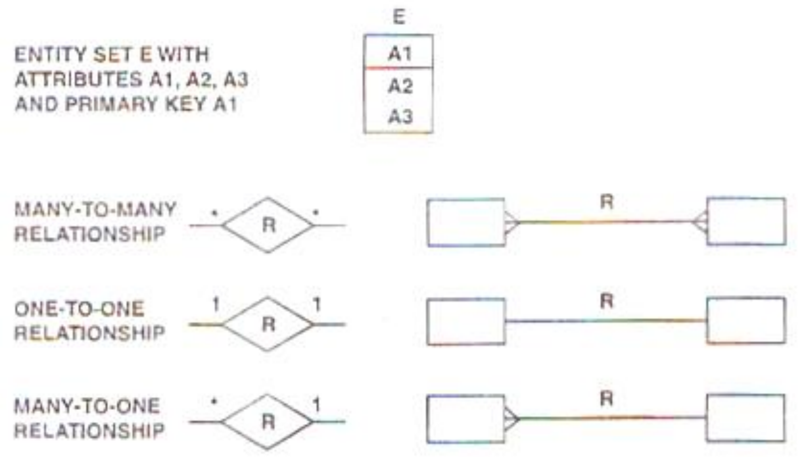
There are conventions for representing the entities and attributes in the E-R diagram.

- The entities are represented by a rectangular box with the name of the entity in the box.
- An attributes is shown as an ellipse attached to a relevant entity by a line and labeled with the attributed name.
- The entity name is written in uppercase where as the attributes name is written in lowercase.
- The primary keys (key attributes) are underlined.
- The attributes are connected using lines to the entities. If the attributes is simple or single valued a single lie is used.
- If the attributes is derived a dotted line is used,
- If it is multi-valued than double lines are used.
- If the attributed is composite, its components attributes are shown as ellipses emanating from the composite attribute.




RELATIONSHIP CARDINALITY
MANDATORY ONE
MANDATORY MANY
OPTIONAL ONE
OPTIONAL MANY


E
ENTITY SET
A
ATTRIBUTE
E
WEAK ENTITY SET
A
MULTIVALUED ATTRIBUTE
R
RELATIONSHIP SET
A
DERIVED ATTRIBUTE
R
IDENTIFYING RELATIONSHIP SET FOR WEAK ENTITY SET
R
E
TOTAL PARTICIPATION OF ENTITY SET IN RELATIONSHIP
A
PRIMARY KEY
A
DISCRIMINATING ATTRIBUTE OF WEAK ENTITY SET
R
MANY-TO-MANY RELATIONSHIP
R
MANY-TO-ONE RELATIONSHIP
R
ONE-TO-ONE RELATIONSHIP
l..h.
R
E
CARDINALITY LIMITS
role name
R
E
ROLE INDICATOR
ISA
ISA (SPECIALIZATION OR GENERALIZATION)
ISA
TOTAL GENERALIZATION
ISA
DISJOINT
DISJOINT GENERALIZATION


E
ENTITY SET E WITH
ATTRIBUTES A1, A2, A3
AND PRIMARY KEY A1
A1
A2
A3
MANY-TO-MANY
RELATIONSHIP
*
R
*
R
ONE-TO-ONE
RELATIONSHIP
1
R
1
R
MANY-TO-ONE
RELATIONSHIP
*
R
1
R