Unit-I: Overview of Database Management System

Data: ***Data*** are the raw alphanumeric values obtained through different acquisition methods. Data in their simplest form consist of **raw alphanumeric values**.

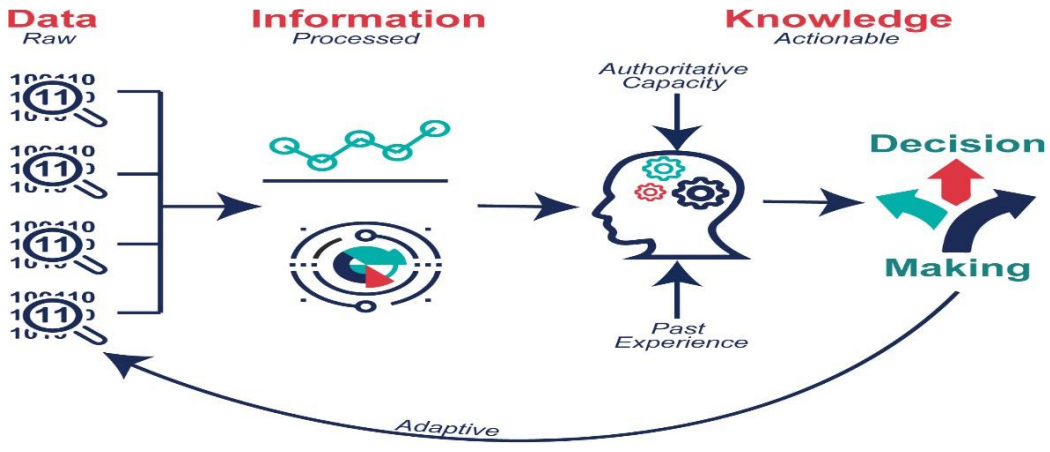
- Data is a collection of a distinct small unit of information. It can be used in a variety of forms like text, numbers, media, bytes, etc. it can be stored in pieces of paper or electronic memory, etc.
- Word 'Data' is originated from the word 'datum' that means 'single piece of information. It is plural of the word datum.
- In computing, Data is information that can be translated into a form for efficient movement and processing. Data is interchangeable.

Information: Information is created when data are processed, organized, or structured to provide context and meaning. Information is essentially **processed data**.

- the collected facts and data about a particular subject
- A telephone service that supplies telephone numbers to the public on request.
- computer data that has been organized and presented in a systematic fashion to clarify the underlying meaning

Knowledge: Knowledge is what we know. Knowledge is unique to each individual and is the accumulation of past experience and insight that shapes the lens by which we interpret, and assign meaning to, information.

- Knowledge is a result in action, an individual must have the authority and capacity to make and implement a decision. Knowledge (and authority) are needed to produce **actionable information** that can lead to impact**.**



Note: The flow from data to information and knowledge is not uni-directional. The knowledge gained may reveal redundancies or gaps in the data collected. As a result, an actionable insight may be to change the data collected, or how those data are converted into information, to better meet user needs.

Data	Information	Knowledge
Is objective	Should be objective	Is subjective
Has no meaning	Has a meaning	Has meaning for a specific purpose
Is unprocessed	Is processed	Is processed and understood
Is quantifiable, there can be data overload	Is quantifiable, there can be information overload	Is not quantifiable, there is no knowledge overload

Figure: **Characteristics of data, information, and knowledge**

Database:

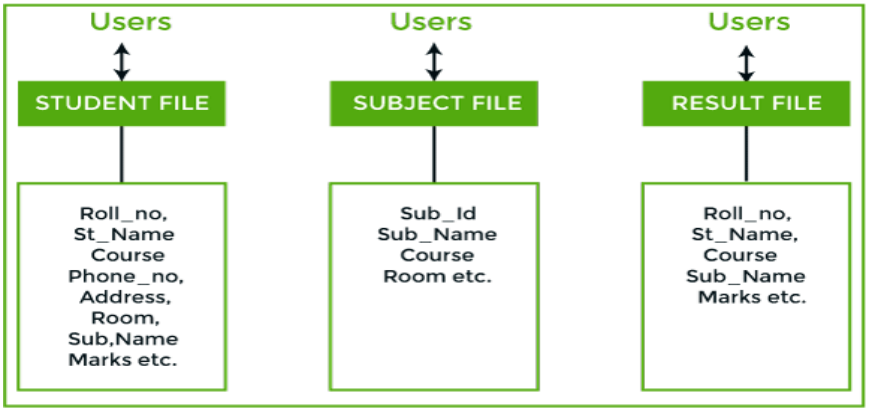
- A database is a systematic collection of data. They support electronic storage and manipulation of data. Databases make data management easy.
- A database is an organized collection of structured information, or data, typically stored electronically in a computer system.
- A database is usually controlled by a database management system (DBMS).

For example:

- An **online telephone directory** uses a database to store data of people, phone numbers, and other contact details.
- Your **electricity service provider** uses a database to manage billing, client-related issues, handle fault data, etc.

File Based Management System:

- File based systems were an early attempt to computerize the manual system.
- It is also called a traditional based approach in which a decentralized approach was taken where each department stored and controlled its own data with the help of a data processing specialist.
- The main role of a data processing specialist was to create the necessary computer file structures, and also manage the data within structures and design some application programs that create reports based on file data.
- C/C++ and COBOL languages were used to design the files.



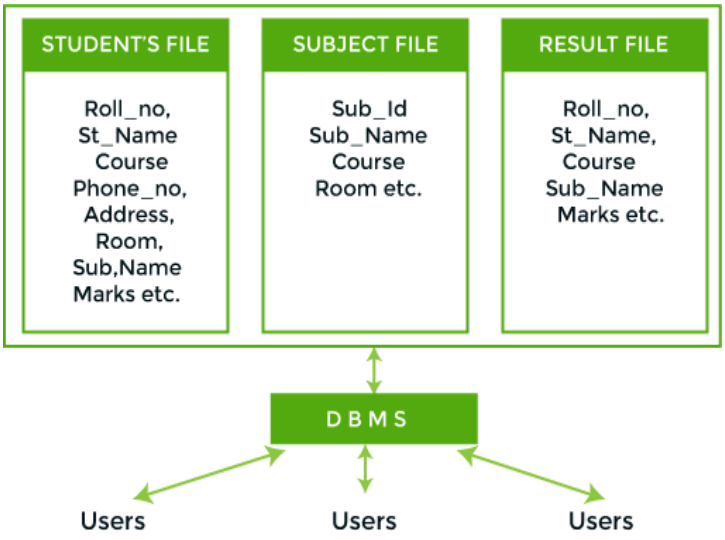
In the above figure:

Consider an example of a student's file system. The student file will contain information regarding the student (i.e. roll no, student name, course etc.). Similarly, we have a subject file that contains information about the subject and the result file which contains the information regarding the result.

Some fields are duplicated in more than one file, which leads to data redundancy. So to overcome this problem, we need to create a centralized system, i.e. DBMS approach.

DBMS (Database Management System**):** A database approach is a well-organized collection of data that are related in a meaningful way which can be accessed by different users but stored only once in a system. The various operations performed by the DBMS system are: Insertion, deletion, selection, sorting etc.

- The DBMS was introduced during 1960's to store any data.
- A DBMS is a software used to store and manage data.
- It also offers manipulation of the data like insertion, deletion, and updating of the data.
- DBMS system also performs the functions like defining, creating, revising and controlling the database.
- It is specially designed to create and maintain data and enable the individual business application to extract the desired data.



In the above figure,

In the above figure, duplication of data is reduced due to centralization of data.

RDBMS (Relational Database Management System**):**

- It came into existence during 1970's.
- Relational Database Management System (RDBMS) is an advanced version of a DBMS system.
- RDBMS system also allows the organization to access data more efficiently than DBMS.
- RDBMS is a software system which is used to store only data which need to be stored in the form of tables.
- In this kind of system, data is managed and stored in rows and columns which is known as tuples and attributes.
- RDBMS is a powerful data management system and is widely used across the world.

Comparison Between File Based and Database System

File Based System	DBMS
A file system is a software that manages and organizes the files in a storage medium. It controls how data is stored and retrieved.	DBMS or Database Management System is a software application. It is used for accessing, creating, and managing databases.
The file system provides the details of data representation and storage of data.	DBMS gives an abstract view of data that hides the details
Storing and retrieving of data can't be done efficiently in a file system.	DBMS is efficient to use as there are a wide variety of methods to store and retrieve data.
It does not offer data recovery processes.	There is a backup recovery for data in DBMS.
The file system doesn't have a crash recovery mechanism.	DBMS provides a crash recovery mechanism
Protecting a file system is very difficult.	DBMS offers good protection mechanism.
In a file management system, the redundancy of data is greater.	The redundancy of data is low in the DBMS system.
Data inconsistency is higher in the file system.	Data inconsistency is low in a database management system.
The file system offers lesser security.	Database Management System offers high security.
File System allows you to stores the data as isolated data files and entities.	Database Management System stores data as well as defined constraints and interrelation.
Not provide support for complicated transactions.	Easy to implement complicated transactions.
The centralization process is hard in File Management System.	Centralization is easy to achieve in the DBMS system.
It doesn't offer backup and recovery of data if it is lost.	DBMS system provides backup and recovery of data even if it is lost.
There is no efficient query processing in the file system.	You can easily query data in a database using the SQL language.

These system doesn't offer concurrency.	DBMS system provides a concurrency facility.

Advantages of Database Management System (or) Drawbacks of File System

Compared to the File Based Data Management System, Database Management System has many advantages. Some of these advantages are given below:

1. **Reducing Data Redundancy**

   The file based data management systems contained multiple files that were stored in many different locations in a system or even across multiple systems. Because of this, there were sometimes multiple copies of the same file which lead to data redundancy.

   This is prevented in a database as there is a single database and any change in it is reflected immediately. Because of this, there is no chance of encountering duplicate data.

2. **Sharing of Data**

   In a database, the users of the database can share the data among themselves. There are various levels of authorisation to access the data, and consequently the data can only be shared based on the correct authorisation protocols being followed.

   Many remote users can also access the database simultaneously and share the data between themselves.

3. **Data Integrity**

   Data integrity means that the data is accurate and consistent in the database. Data Integrity is very important as there are multiple databases in a DBMS. All of these databases contain data that is visible to multiple users. So it is necessary to ensure that the data is correct and consistent in all the databases and for all the users.

4. **Data Security**

   Data Security is vital concept in a database. Only authorised users should be allowed to access the database and their identity should be authenticated using a username and password. Unauthorised users should not be allowed to access the database under any circumstances as it violates the integrity constraints.

5. **Privacy**

   The privacy rule in a database means only the authorized users can access a database according to its privacy constraints. There are levels of database access and a user can only view the data he is allowed to. For example - In social networking sites, access constraints are different for different accounts a user may want to access.

6. **Backup and Recovery**

   Database Management System automatically takes care of backup and recovery. The users don't need to backup data periodically because this is taken care of by the DBMS. Moreover, it also restores the database after a crash or system failure to its previous condition.

7. **Data Consistency**

   Data consistency is ensured in a database because there is no data redundancy. All data appears consistently across the database and the data is same for all the users viewing the database. Moreover, any changes made to the database are immediately reflected to all the users and there is no data inconsistency.

Data Models (or) Classification of Database Managements Systems

- A Database model defines the logical design and structure of a database and defines how data will be stored, accessed and updated in a database management system.
- Data Model is the modelling of the data description, data semantics, and consistency constraints of the data.
- It provides the conceptual tools for describing the design of a database at each level of data abstraction.
- Though there are many data models being used nowadays but the Relational model is the most widely used model.

Some of the Data Models in DBMS are:

1. Hierarchical Model
2. Network Model
3. Entity-Relationship Model
4. Relational Model
5. Object-Oriented Data Model
6. Object-Relational Data Model

Hierarchical Model

- This database model organises data into a tree-like-structure, with a single root, to which all the other data is linked. The hierarchy starts from the **Root** data, and expands like a tree, adding child nodes to the parent nodes.

Network Model

- This is an extension of the Hierarchical model. In this model data is organised more like a graph, and are allowed to have more than one parent node.
- This database model was used to map many-to-many data relationships.
- This was the most widely used database model, before Relational Model was introduced.

Relational Data Model:

- This type of model designs the data in the form of rows and columns within a table. Tables are also called relations.
- This model was introduced by E.F Codd in 1970, and since then it has been the most widely used database model, infact, we can say the only database model used around the world.
- The relational data model is the widely used model which is primarily used by commercial data processing applications.

Emp_id	Emp_name	Job_name	Salary	Mobile_no	Dep_id	Project_id
AfterA001	John	Engineer	100000	9111037890	2	99
AfterA002	Adam	Analyst	50000	9587569214	3	100
AfterA003	Kande	Manager	890000	7895212355	2	65

EMPLOYEE TABLE

Advantages of Relational Model

- ***Simple:*** This model is simpler as compared to the network and hierarchical model.
- ***Scalable:*** This model can be easily scaled as we can add as many rows and columns we want.
- ***Structural Independence:*** We can make changes in database structure without changing the way to access the data. When we can make changes to the database structure without affecting the capability to DBMS to access the data we can say that structural independence has been achieved.
- ***Design, implementation, maintenance and usage free-***

This model is both data independence and structural independence making the design ,maintenance, administration and usage much easier than the other models.

Adhoc capability-The presence of SQL is the main reason for the popularity of the model. Sql makes the adhoc query a reality.

In SQL the user specify what they wantand leave the details of how to get the information to the database.

Relational database perform the task of translating the query to the technical code required to retrieve information.

Disadvantages of Relational Model

- ***Hardware Overheads:*** For hiding the complexities and making things easier for the user this model requires more powerful hardware computers and data storage devices.
- ***Bad Design:*** As the relational model is very easy to design and use. So the users don't need to know how the data is stored in order to access it. This ease of design can lead to the development of a poor database which would slow down if the database grows.

But all these disadvantages are minor as compared to the advantages of the relational model. These problems can be avoided with the help of proper implementation and organisation.

Entity-Relationship Model

- Entity-Relationship Model or simply ER Model is a high-level data model diagram.
- In this model, we represent the real-world problem in the pictorial form to make it easy for the stakeholders to understand.
- It is also very easy for the developers to understand the system by just looking at the ER diagram.

Object-Oriented Data Model

- The real-world problems are more closely represented through the object-oriented data model.
- In this model, both the data and relationship are present in a single structure known as an object.

Object-Relational Model

- As the name suggests it is a combination of both the relational model and the object- oriented model.
- This model was built to fill the gap between object-oriented model and the relational model.
- We can have many advanced features like we can make complex data types according to our requirements using the existing data types.
- The problem with this model is that this can get complex and difficult to handle. So, proper understanding of this model is required.

--RDBMS Terminology – The Relational Data Structure – Relational Data Integrity – Codd's Rules – Database Architecture and Data Modelling: Conceptual, Physical and Logical Models--- Refer Notes given in class

E-R Modelling

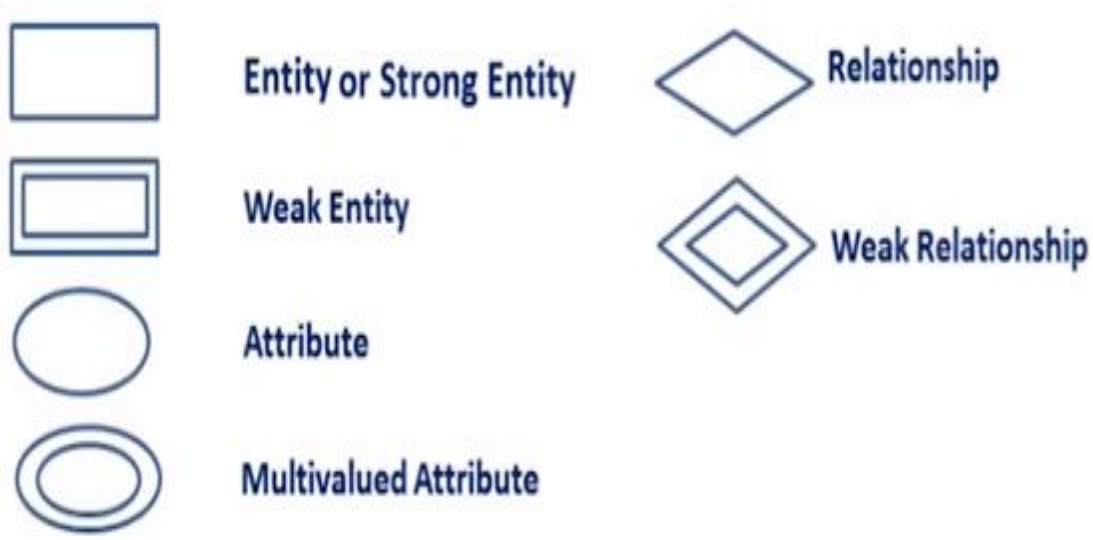
- ER model stands for an **Entity-Relationship** model. It is a high-level data model.
- The Entity Relationship model was proposed by **Peter Chen** in 1976.
- ER model is a logical representation of an enterprise data. ER model is a diagrammatic representation of logical structure of database.
- E-R model describes relationship among entities and attributes.
- Entity Relationship Diagrams are the best tools to communicate within the entire system.
- These diagrams are the graphical representation of the flow of data and information.
- These diagrams are most commonly used in business organizations to make data travel easy.
- This conceptual database model is an effective way of communicating with the

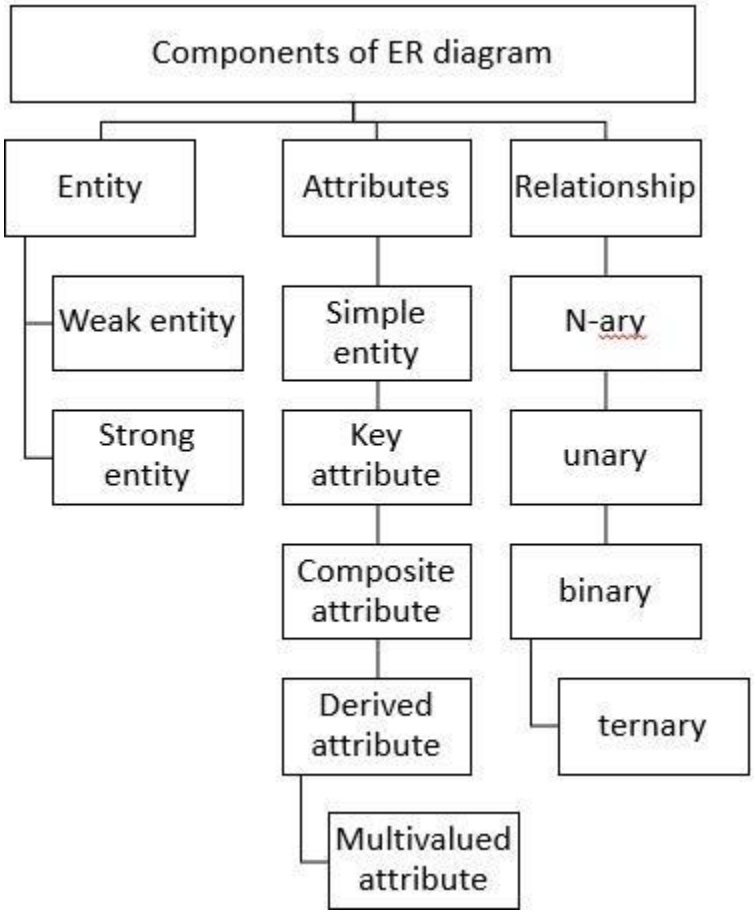
individuals at all the levels.

- The most common use of this diagram is to present the relation of the various tables present in a database.

Following are the main components and its symbols in ER Diagrams:

- **Rectangles:** This Entity Relationship Diagram symbol represents entity types
- **Ellipses :** Symbol represent attributes
- **Diamonds:** This symbol represents relationship types
- **Lines:** It links attributes to entity types and entity types with other relationship types
- **Primary key:** attributes are underlined
- **Double Ellipses:** Represent multi-valued attributes





1. Entity

It may be an object, person, place or event that stores data in a database. In E-R diagram an entity is represented in rectangle form. For example, students, employees, managers, etc.

The entity is pictorially depicted as follows:

Entity set

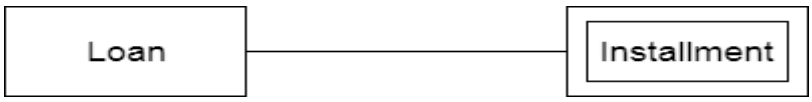
It is a collection of entities of the same type which share similar properties. For example, a group of students in a college and students are an entity set.

Entity is characterised into two types as follows:

a. Strong entity set
b. Weak entity set

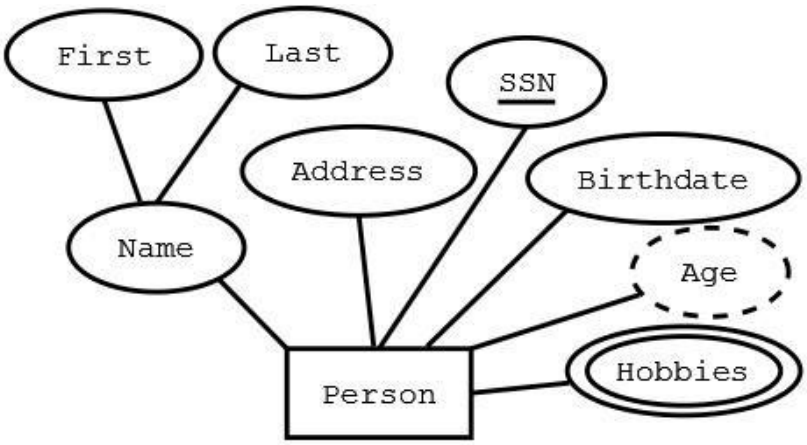
a. **Strong entity set:** The entity types which consist of key attributes or if there are enough attributes for forming a primary key attribute are called a strong entity set. It is represented by a single rectangle.

b. **Weak entity set:** An entity does not have a primary key attribute and depends on another strong entity via foreign key attribute. It is represented by a double rectangle.



2. Attributes

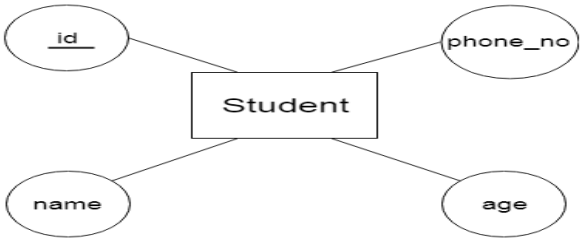
These are the data characteristics of entities or data elements and data fields.



Types of attributes

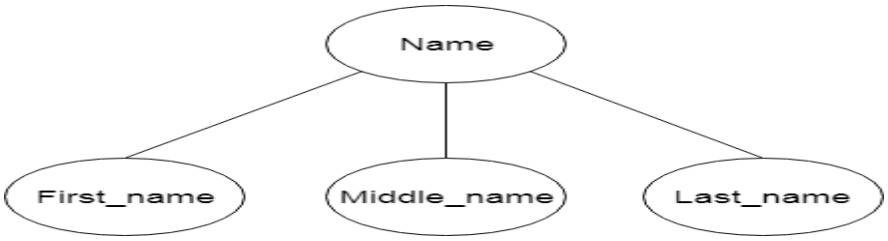
The types of attributes in the Entity Relationship (ER) model are as follows:

1. **Single value attribute** − these attributes contain a single value. For example, age, salary etc.
2. **Key Attributes**- The key attribute is used to represent the main characteristics of an entity. It represents a primary key. The key attribute is represented by an ellipse with the text underlined.

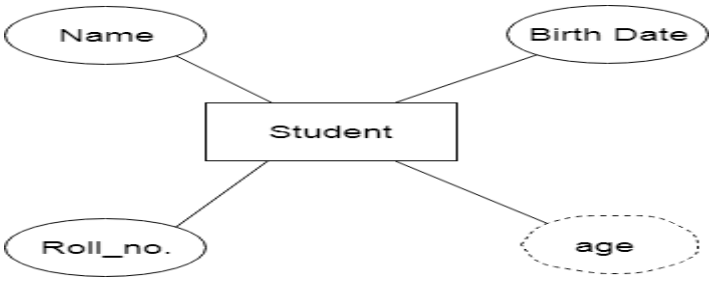


3. **Multivalued attribute** − they contain more than one value of a single entity. For example, phone numbers, Email_Ids, etc.

4. **Composite attribute** − the attributes which can be further divided. For example, **Name** consists of First name, Middle name, last name



5. **Derived attribute** − the attribute that can be derived from other attributes. For example, age can be derived based on DoB.



3. Relationships

A relationship is used to describe the relation between entities. Diamond or rhombus is used to represent the relationship.

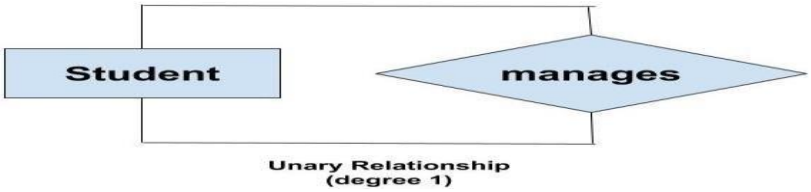
Degree of Relationship:

A relationship where a number of different entities set participate is called a degree of a relationship.

It is categorised into the following:

1. **Unary Relationship:** A unary relationship exists when both the participating entity type are the same. When such a relationship is present we say that the degree of relationship is 1.

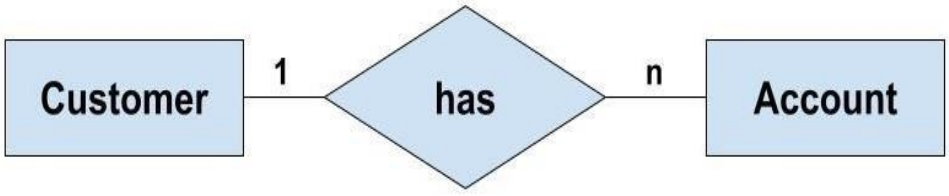
   For example, suppose in a classroom, we have many students who belong to a particular club-like dance club, basketball club etc. and some of them are club leads. So, a particular group of student is managed by their respective club lead. Here, the group is formed from students and also, the club leads are chosen from students. So, the 'Student' is the only entity participating here. We can represent this relationship using the E-R diagram as follows:



Unary Relationship
(degree 1)

2. **Binary Relationship:** A binary relationship exists when exactly two entity type participates. When such a relationship is present we say that the degree is 2. This is the most common degree of relationship. It is easy to deal with such relationship as these can be easily converted into relational tables.

   For example, we have two entity type 'Customer' and 'Account' where each 'Customer' has an 'Account' which stores the account details of the 'Customer'. Since we have two entity types participating we call it a binary relationship. Also, one 'Customer' can have many 'Account' but each 'Account' should belong to only one 'Customer'. We can say that it is a one-to-many binary relationship.

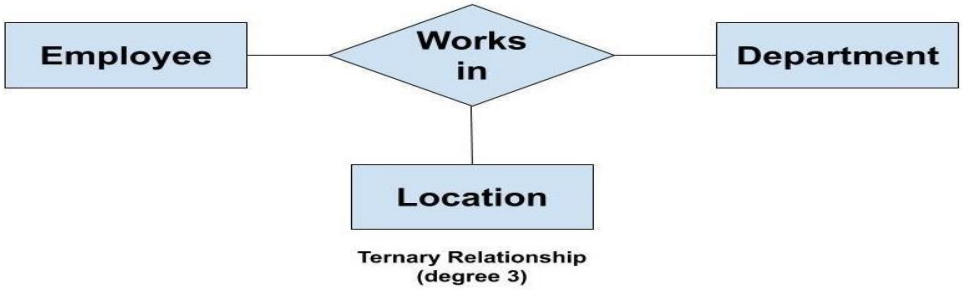


Binary Relationship
(degree 2)

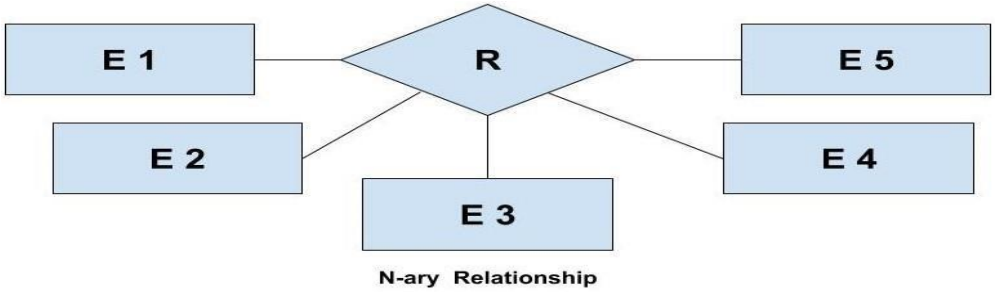
3. **Ternary Relationship:** A ternary relationship exists when exactly three entity type participates. When such a relationship is present we say that the degree is 3. As the number of entity increases in the relationship, it becomes complex to convert them into relational tables.

   For example, we have three entity type 'Employee', 'Department' and 'Location'. The relationship between these entities are defined as an employee works in a department,

an employee works at a particular location. So, we can see we have three entities participating in a relationship so it is a ternary relationship. The degree of this relation is 3.



4. **n-ary Relationship:** An N-ary relationship exists when 'n' number of entities are participating. So, any number of entities can participate in a relationship. There is no limitation to the maximum number of entities that can participate.



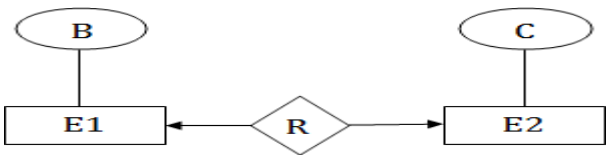
Mapping Constraints or Connectivity

- A mapping constraint is a data constraint that expresses the number of entities to which another entity can be related via a relationship set.
- It is most useful in describing the relationship sets that involve more than two entity sets.

For binary relationship set R on an entity set A and B, there are four possible mapping cardinalities.

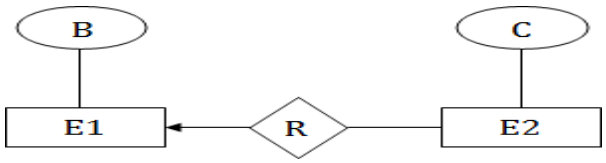
a. One-to-One (1:1)
b. One-to-Many (1:M)
c. Many-to-One (M:1)
d. Many-to-Many (M:M)

a. One-to-One Relationship: When only one instance of an entity is associated with the relationship, then it is known as one to one relationship.



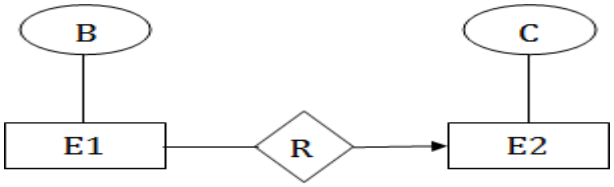
For example: A female can marry to one male, and a male can marry to one female.

b. One-to-many relationship: When only one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then this is known as a one-to-many relationship.



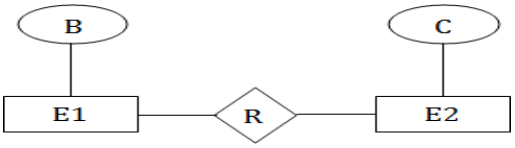
For example, Scientist can invent many inventions, but the invention is done by the only specific scientist.

c. Many-to-one relationship: When more than one instance of the entity on the left, and only one instance of an entity on the right associates with the relationship then it is known as a many-to-one relationship.



For example, Student enrolls for only one course, but a course can have many students.

d. Many-to-many relationship: When more than one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then it is known as a many-to-many relationship.



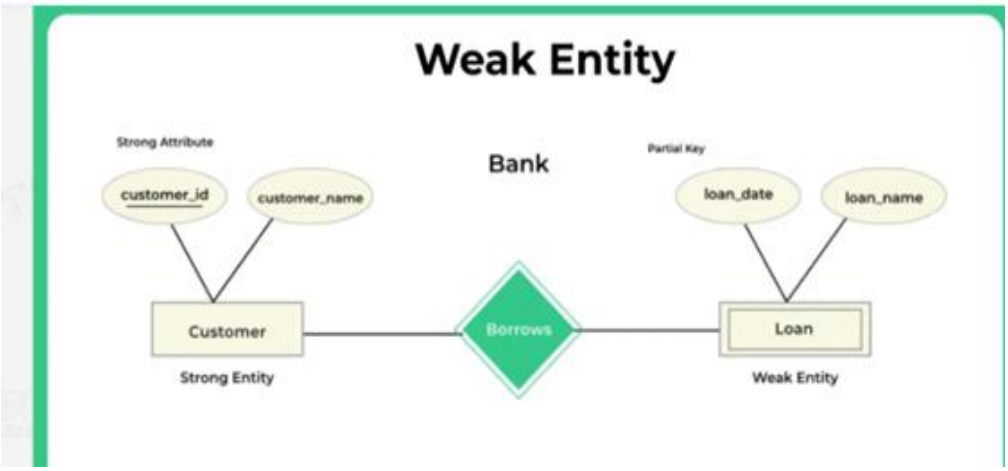
For example, Employee can assign by many projects and project can have many employees.

Dependency

Entities are classified as strong and weak entity

Representation:

- A **single rectangle** is used to represent strong entities.
- A **single diamond** is used to represent the relationship between two strong entities.
- A **double rectangle** is used for representing a **weak entity set**
- The **double diamond** symbol is used for representing the **relationship between a strong entity and a weak entity** which is known as identifying relationship



A loan entity can not be created for a customer if the customer doesn't exist

Comparison between Strong and Weak Entity Set

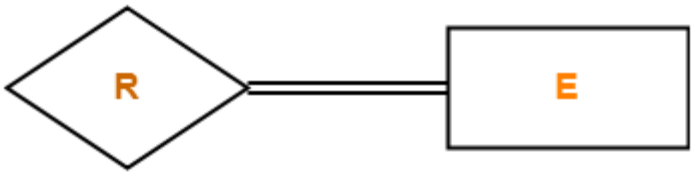
Strong Entity Set	Weak Entity Set
1. Strong entity set always has a primary key.	1. It does not have enough attributes to build a primary key.
2. It is represented by a rectangle symbol.	2. It is represented by a double rectangle symbol.
3. It contains a Primary key represented by the underline symbol.	3. It contains a Partial Key which is represented by a dashed underline symbol.

4. The member of a strong entity set is called as dominant entity set.	4. The member of a weak entity set called as a subordinate entity set.
5. Primary Key is one of its attributes which helps to identify its member.	5. In a weak entity set, it is a combination of primary key and partial key of the strong entity set.
6. In the ER diagram the relationship between two strong entity set shown by using a diamond symbol.	6. The relationship between one strong and a weak entity set shown by using the double diamond symbol.
7. The connecting line of the strong entity set with the relationship is single.	7. The line connecting the weak entity set for identifying relationship is double.

Participation
There are two ways
1.Total(Mandatory)
2.Partiall(optional)

1. Total Participation-

1. **It specifies that each entity in the entity set must compulsorily participate in at least one relationship instance in that relationship set.That is why, it is also called as mandatory participation.**
2. **Total participation is represented using a double line between the entity set and relationship set.**



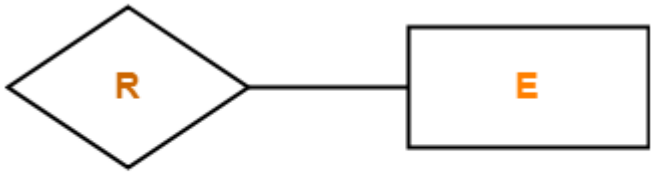
Total Participation

Example-

Here,
Double line between the entity set "Student" and relationship set "Enrolled in" signifies total participation. It specifies that each student must be enrolled in at least one course.

2. Partial Participation-

1. **It specifies that each entity in the entity set may or may not participate in the relationship instance in that relationship set. That is why, it is also called as optional participation.**
2. **Partial participation is represented using a single line between the entity set and relationship set.**



Partial Participation

Example-

Here,
Single line between the entity set "Course" and relationship set "Enrolled in" signifies partial participation. It specifies that there might exist some courses for which no enrollments are made

Composite Entities
A Composite Entity is defined as an entity that represents the relationship between two or more other entities.
It is created to simplify complex relationships by acting as an intermediary entity, storing specific relationship data if needed.
Composite entity in an ER model refers to an entity that is created specifically to represent a many-to-many relationship between two other entities, essentially acting as a bridge between them and capturing additional details about the connection.

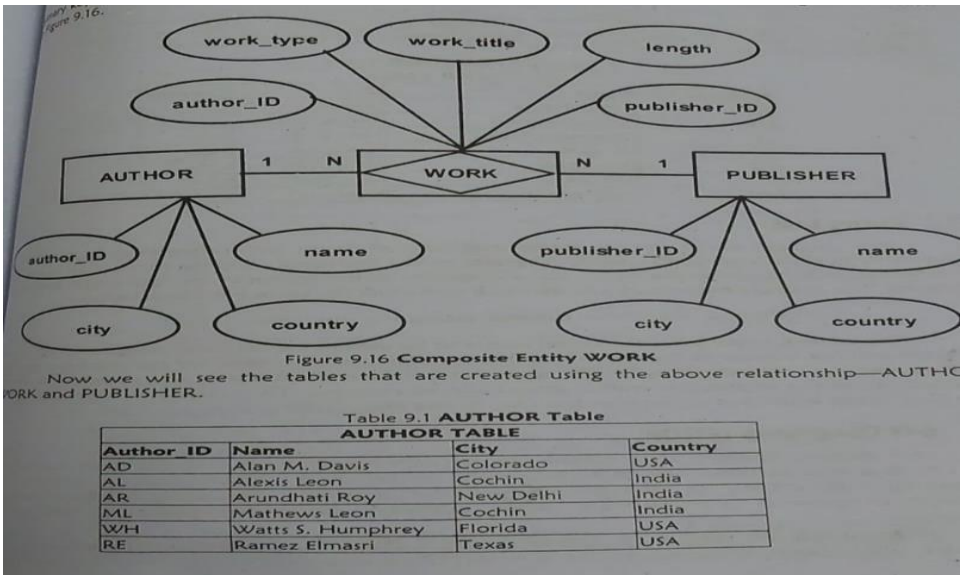


Figure 9.16 **Composite Entity WORK**

Now we will see the tables that are created using the above relationship—AUTHOR, WORK and PUBLISHER.

Table 9.1 **AUTHOR Table**

AUTHOR TABLE

Author_ID	Name	City	Country
AD	Alan M. Davis	Colorado	USA
AL	Alexis Leon	Cochin	India
AR	Arundhati Roy	New Delhi	India
ML	Mathews Leon	Cochin	India
WH	Watts S. Humphrey	Florida	USA
RE	Ramez Elmasri	Texas	USA

In an ER model for a database containing "Author" and "Publisher" entities, a "Book" entity would be considered a composite entity, as it represents the relationship between an author and a publisher, effectively acting as a bridge between the two entities in a many-to-many relationship.
Explanation:

- **Many-to-Many Relationship:**

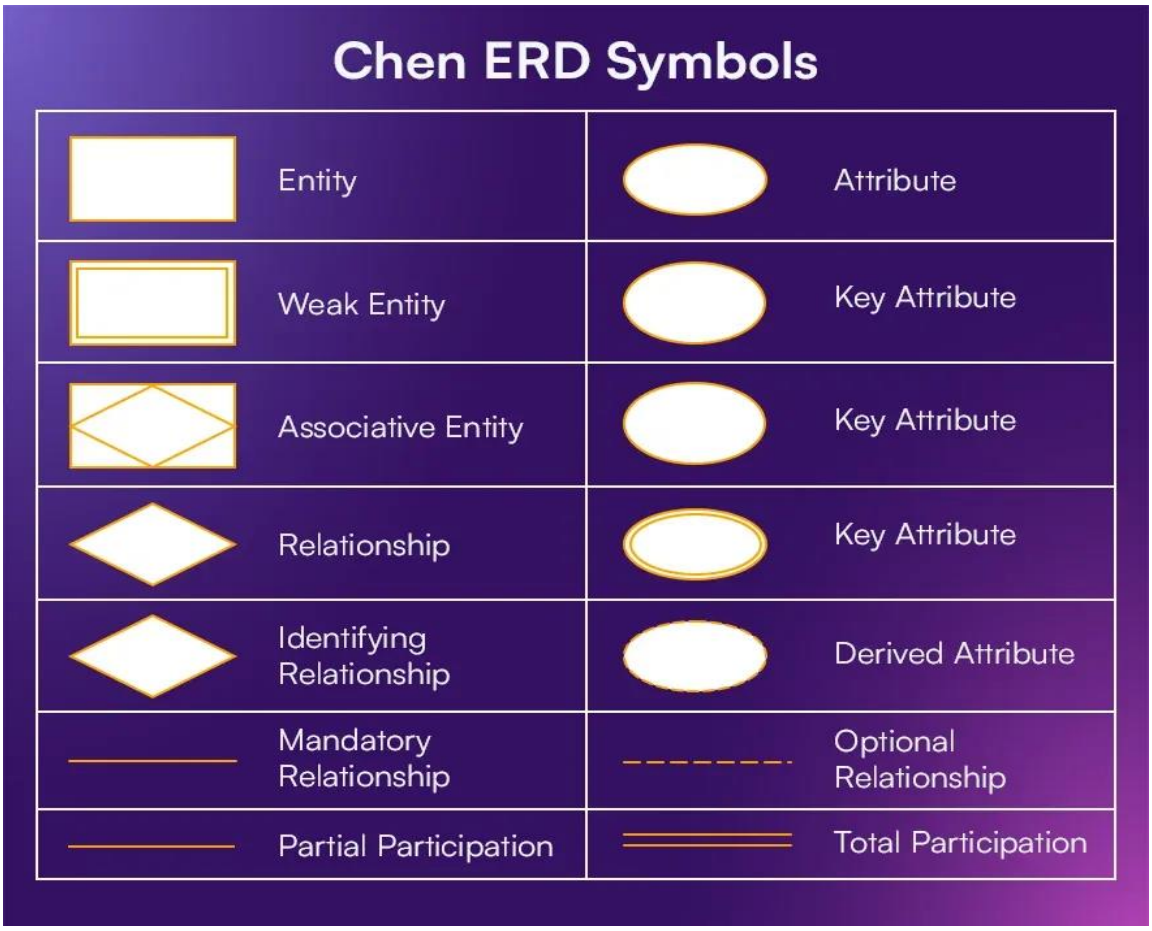
An author can write multiple books, and a publisher can publish multiple books, creating a many-to-many relationship between "Author" and "Publisher."

- **Composite Entity Function:**

The "Book" entity acts as a bridge by holding the necessary foreign keys from both "Author" and

"Publisher" tables, allowing you to link specific authors to specific published books.

ER model Symbols



Unit III