|	
COURSE TITLE:	SYSTEM ANALYSIS AND DESIGN
PAPER CODE:	CSE 4205
LECTURER:	MASESE CHUMA
TOPIC:	SYSTEM DESIGN
CHAPTER:	Five

Introduction

System design is the phase that bridges the gap between problem domain and the existing system in a manageable way. This phase focuses on the solution domain, i.e. "*how to implement?*"

It is the phase where the SRS document is converted into a format that can be implemented and decides how the system will operate.

In this phase, the complex activity of system development is divided into several smaller sub-activities, which coordinate with each other to achieve the main objective of system development.
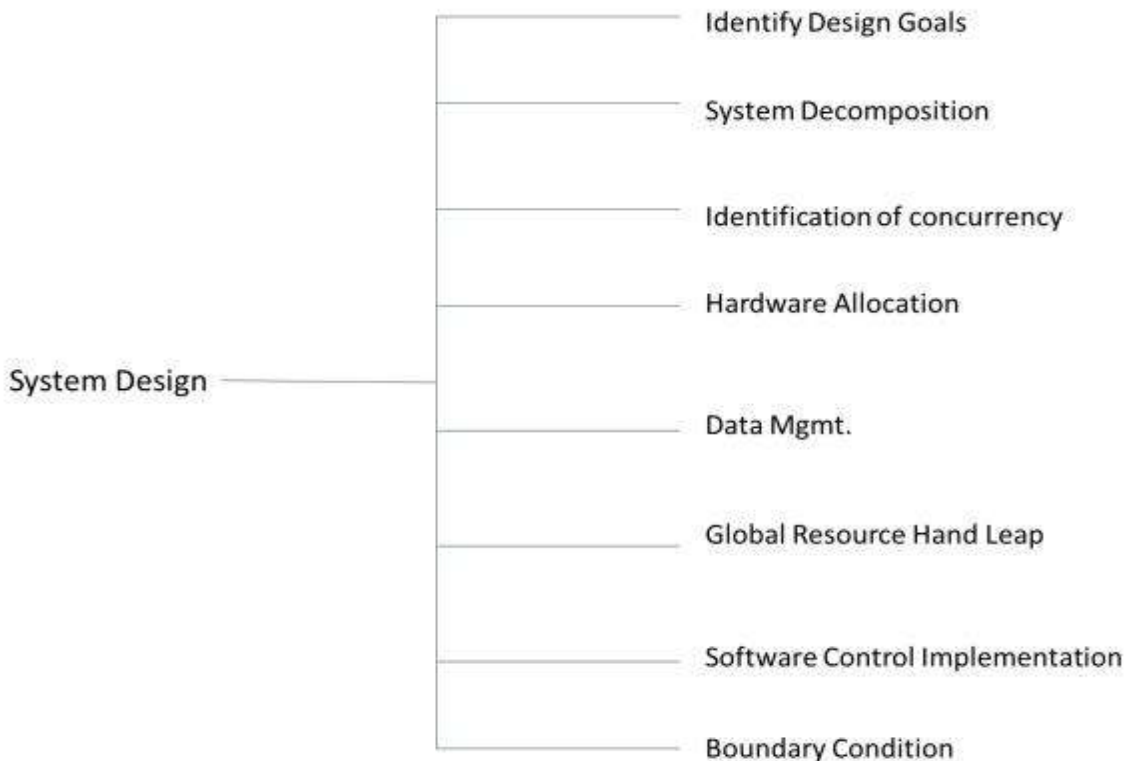
Inputs to System Design

System design takes the following inputs:

- Statement of work

- Requirement determination plan

- Current situation analysis

- Proposed system requirements including a conceptual data model, modified

- DFDs, and Metadata (data about data).

Outputs for System Design

System design gives the following outputs:

- Infrastructure and organizational changes for the proposed system.

- A data schema, often a relational schema.

- Metadata to define the tables/files and columns/data-items.

- A function hierarchy diagram or web page map that graphically

- describes the program structure.

- Actual or pseudocode for each module in the program.

- A prototype for the proposed system.

Types of System Design

Logical Design

Logical design pertains to an abstract representation of the data flow, inputs, and outputs of the system. It describes the inputs (sources), outputs (destinations), databases (data stores), procedures (data flows) all in a format that meets the user requirements.

While preparing the logical design of a system, the system analyst specifies the user needs at level of detail that virtually determines the information flow into and out of the system and the required data sources. Data flow diagram, E-R diagram modeling are used.

Physical Design

Physical design relates to the actual input and output processes of the system. It focuses on how data is entered into a system, verified, processed, and displayed as output.

It produces the working system by defining the design specification that specifies exactly what the candidate system does. It is concerned with user interface design, process design, and data design.

It consists of the following steps:

1. Specifying the input/output media, designing the database, and specifying backup procedures.
2. Planning system implementation.
3. Devising a test and implementation plan, and specifying any new hardware and software.
4. Updating costs, benefits, conversion dates, and system constraints.

Architectural Design

- It is also known as high level design that focuses on the design of system architecture.
- It describes the structure and behavior of the system. It defines the structure and relationship between various modules of system development process.

Detailed Design

It follows Architectural design and focuses on development of each module.

Conceptual Data Modeling

It is representation of organizational data which includes all the major entities and relationship. System analysts develop a conceptual data model for the current system that supports the scope and requirement for the proposed system.

The main aim of conceptual data modeling is to capture as much meaning of data as possible. Most organization today use conceptual data modeling using E-R model which uses special notation to represent as much meaning about data as possible.

Entity Relationship Model

It is a technique used in database design that helps describe the relationship between various entities of an organization.

Terms used in E-R model

ENTITY: It specifies distinct real world items in an application. For example:
vendor, item, student, course, teachers, etc.

RELATIONSHIP: They are the meaningful dependencies between entities.
For example, vendor supplies items, teacher teaches courses, then supplies and course are relationship.

ATTRIBUTES: It specifies the properties of relationships. For example, vendor code, student name.

Symbols used in E-R model and their respective meanings:

The following table shows the symbols used in E-R model and their significance:

Entity-Relationship Model

Entity-Relationship (ER) Model is based on the notion of real-world entities and relationships among them. While formulating real-world scenario into the database model, the ER Model creates entity set, relationship set, general attributes and constraints.

ER Model is best used for the conceptual design of a database.

ER Model is based on –

- **Entities** and their *attributes.*

- **Relationships** among entities.

These concepts are explained below.
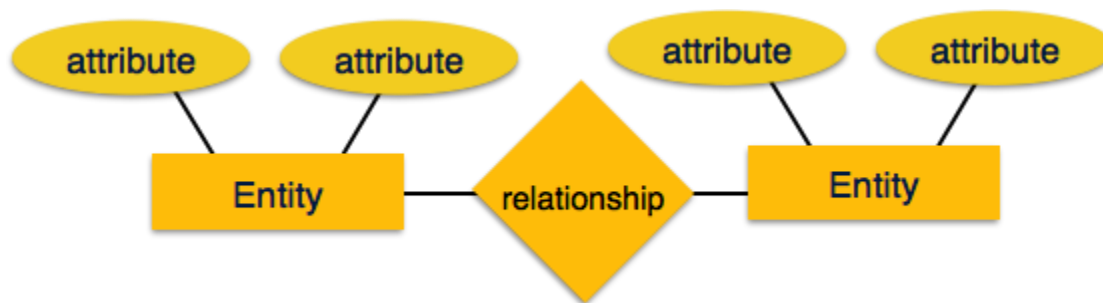
- **Entity** – An entity in an ER Model is a real-world entity having properties called **attributes**. Every **attribute** is defined by its set of values called **domain**. For example, in a school database, a student is considered as an entity. Student has various attributes like name, age, class, etc.

- **Relationship** – The logical association among entities is called *relationship*. Relationships are mapped with entities in various ways. Mapping cardinalities define the number of association between two entities.
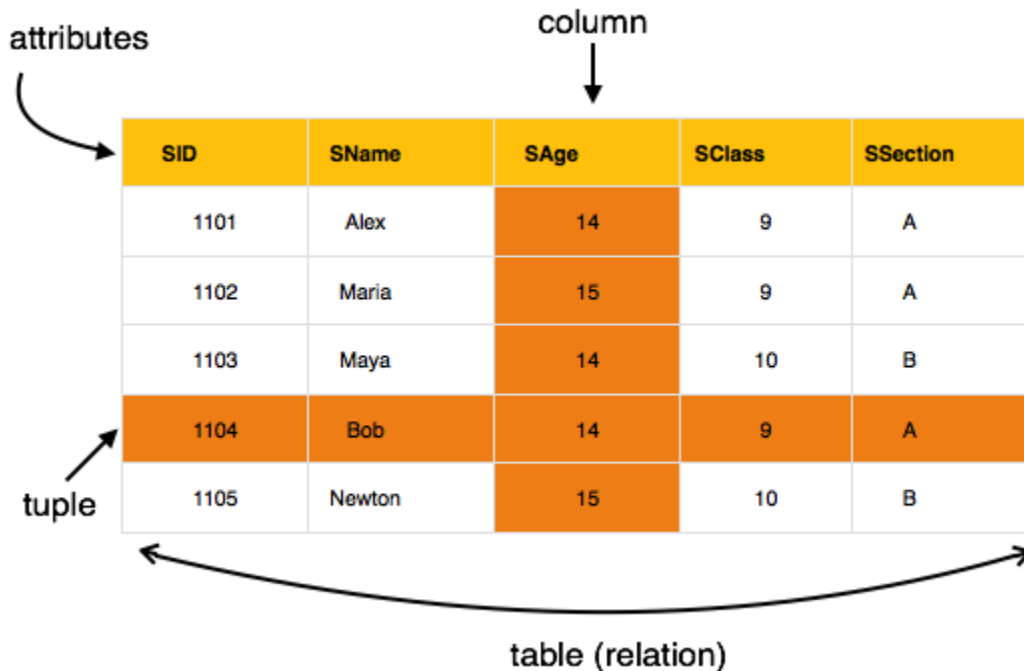
  Mapping cardinalities –

    o   one to one

    o   one to many

    o   many to one

o   many to many

Relational Model

The most popular data model in DBMS is the Relational Model. It is more scientific a model than others.

This model is based on first-order predicate logic and defines a table as an **n-ary relation**.

The main highlights of this model are −

- Data is stored in tables called **relations**.

- Relations can be normalized.

- In normalized relations, values saved are atomic values.

- Each row in a relation contains a unique value.

- Each column in a relation contains values from a same domain.

Entity

Entities are represented by means of rectangles. Rectangles are named with the entity set they represent.

Student   Teacher   Projects

Attributes

Attributes are the properties of entities. Attributes are represented by means of ellipses. Every ellipse represents one attribute and is directly connected to its entity (rectangle).

Name   BirthDate

Student

Roll_No.
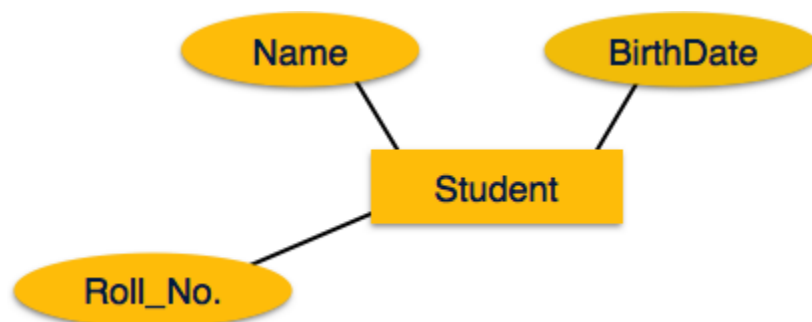
If the attributes are **composite**, they are further divided in a tree like structure. Every node is then connected to its attribute. That is, composite attributes are represented by ellipses that are connected with an ellipse.

LastName

FirstName

Name   BirthDate

Student

Roll_No.
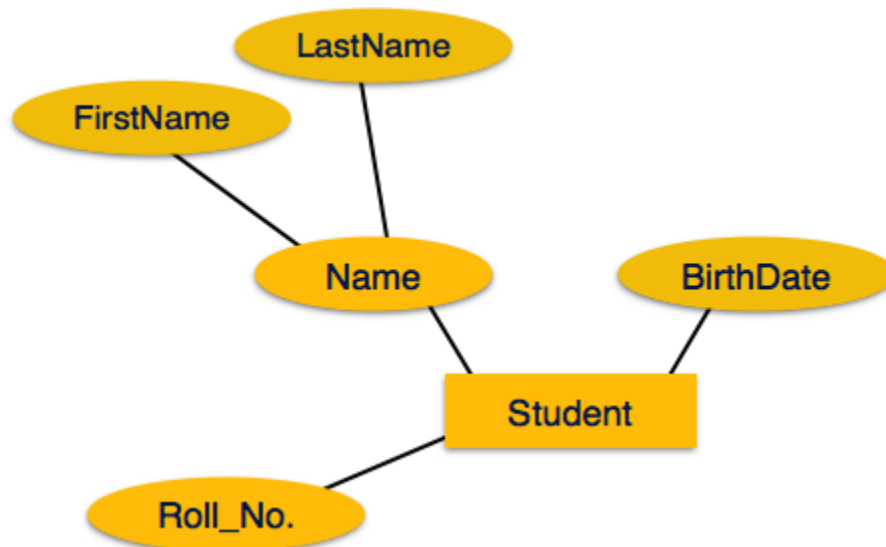
Multivalued attributes are depicted by double ellipse.
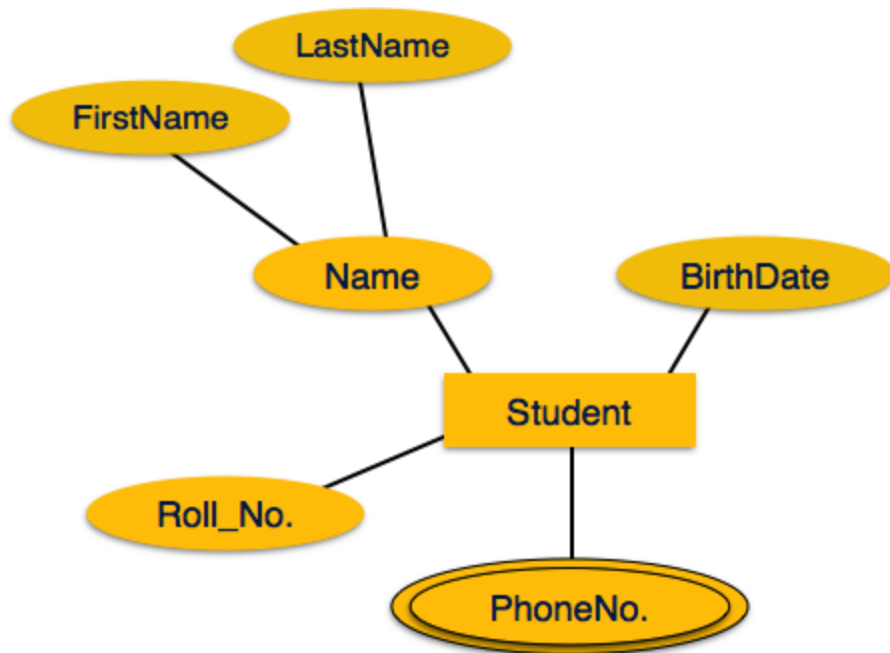
**Derived** attributes are depicted by dashed ellipse.
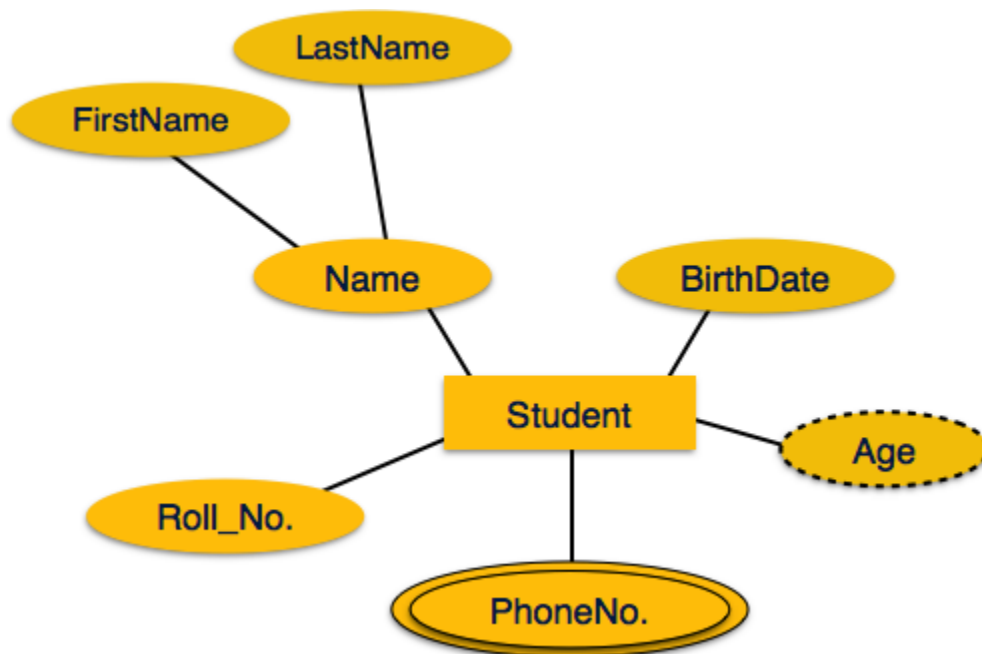
Relationship

Relationships are represented by diamond-shaped box. Name of the relationship is written inside the diamond-box. All the entities (rectangles) participating in a relationship, are connected to it by a line.

Binary Relationship and Cardinality

A relationship where two entities are participating is called a **binary relationship**. Cardinality is the number of instance of an entity from a relation that can be associated with the relation.

- **One-to-one** – When only one instance of an entity is associated with the relationship, it is marked as '1:1'. The following image reflects that only one instance of each entity should be associated with the relationship. It depicts one-to-one relationship.
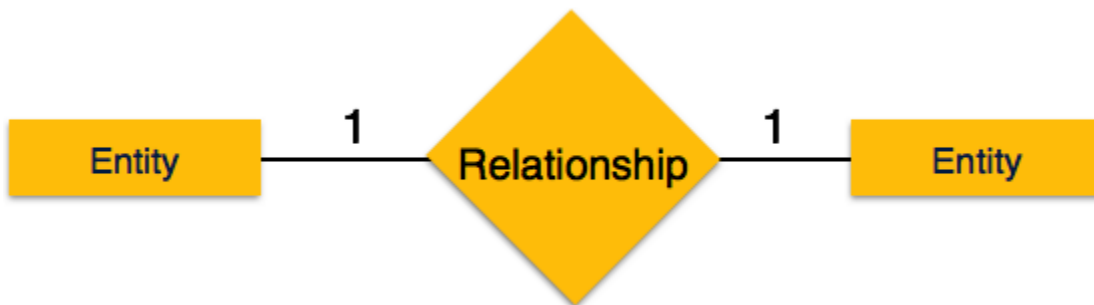

- **One-to-many** – When more than one instance of an entity is associated with a relationship, it is marked as '1:N'. The following image reflects that only one instance of entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts one-to-many relationship.
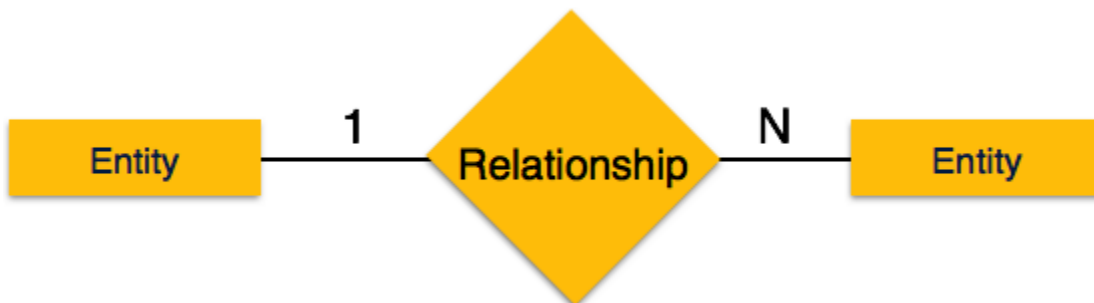
- **Many-to-one** – When more than one instance of entity is associated with the relationship, it is marked as 'N:1'. The following image reflects that more than one instance of an entity on the left and only one instance of an entity on the right can be associated with the relationship. It depicts many-to-one relationship.
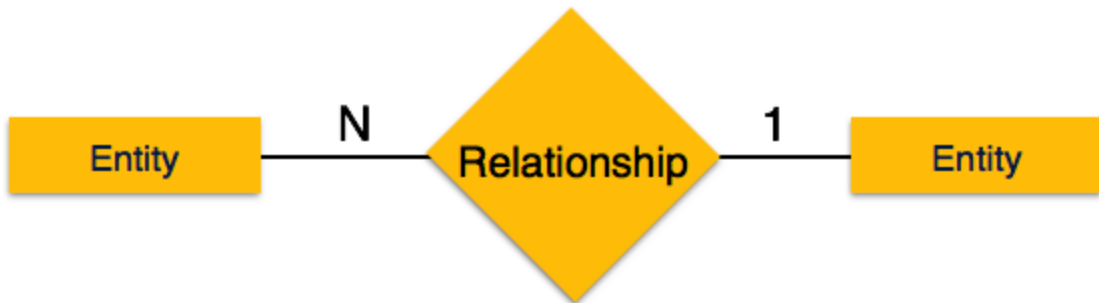
- **Many-to-many** − The following image reflects that more than one instance of an entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts many-to-many relationship.
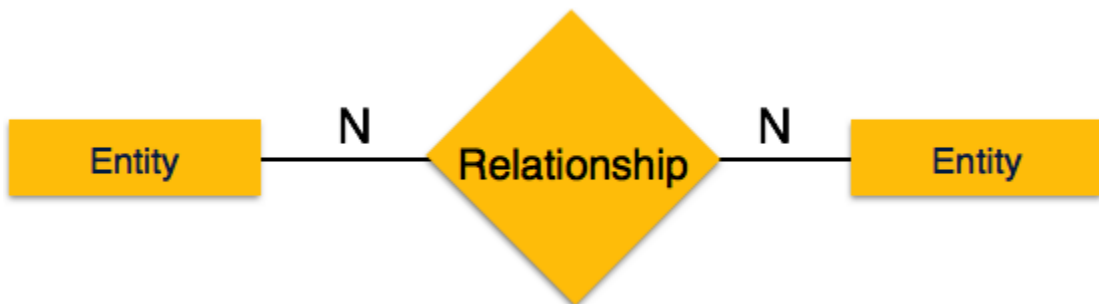
Participation Constraints

- **Total Participation** − Each entity is involved in the relationship. Total participation is represented by double lines.

- **Partial participation** − Not all entities are involved in the relationship. Partial participation is represented by single lines.
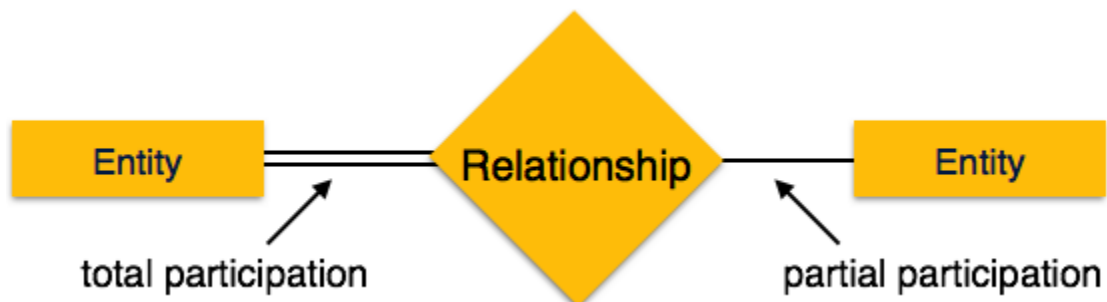
# Entity Relationship Diagram – ER Diagram in DBMS

An **Entity–relationship model (ER model)** describes the structure of a database with the help of a diagram, which is known as **Entity Relationship Diagram (ER Diagram)**. An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

What is an Entity Relationship Diagram (ER Diagram)?

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Lets have a look at a simple ER diagram to understand this concept.

A simple ER Diagram:

In the following diagram we have two entities Student and College and their relationship. The relationship between Student and College is many to one as a college can have many students however a student cannot study in multiple colleges at the same time. Student entity has attributes such as Stu_Id, Stu_Name & Stu_Addr and College entity has attributes such as Col_ID & Col_Name.
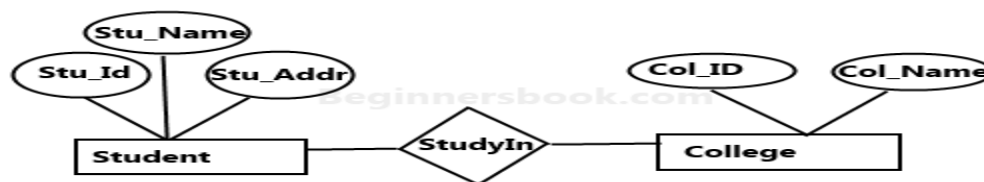
Sample E-R Diagram

Here are the geometric shapes and their meaning in an E-R Diagram. We will discuss these terms in detail in the next section(Components of a ER Diagram) of this guide so don't worry too much about these terms now, just go through them once.

Rectangle: Represents Entity sets.

Ellipses: Attributes

Diamonds: Relationship Set

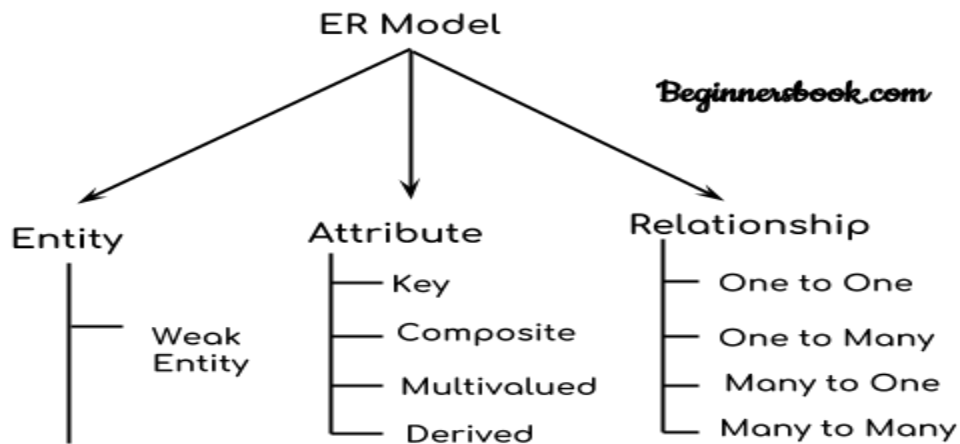**Lines**: They link attributes to Entity Sets and Entity sets to Relationship Set

Double Ellipses: Multivalued Attributes

Dashed Ellipses: Derived Attributes

Double Rectangles: Weak Entity Sets

Double Lines: Total participation of an entity in a relationship set

Components of a ER Diagram

Components of ER Diagram

As shown in the above diagram, an ER diagram has three main components:

1. Entity

2. Attribute

3. Relationship

File Organization

It describes how records are stored within a file.

There are four file organization methods:

Serial: Records are stored in chronological order (in order as they are input or occur).

Examples: Recording of telephone charges, ATM transactions, Telephone queues

Sequential: Records are stored in order based on a key field which contains a value that uniquely identifies a record. **Examples**: Phone directories

Direct (relative): Each record is stored based on a physical address or location on the device. Address is calculated from the value stored in the record's key field.

Randomizing routine or hashing algorithm does the conversion

Indexed: Records can be processed both sequentially and non-sequentially using indexes.

Comparison

	Serial	Sequential	Direct	Index
Type of Access	Batch	Batch	Online	Batch or Online
Data Organization	Serial	Sequentially by key value	No particular order	Sequentially and by index
Flexibility in handling inquiries	No	No	Yes	Yes
Availability of up to date Data	No	No	Yes	Yes
Speed Retrieval	Slow	Slow	Very Fast	Fast
Activity	High	High	Low	High
Volatility	Low	Low	High	High
Example	ATM Transition Queue	Payroll process script billing operation	Online reservation and banking transaction	Customer ordering and billing

File Access

One can access a file using either Sequential Access or Random Access. File Access methods allow computer programs read or write records in a file.

Sequential Access

Every record on the file is processed starting with the first record until End of File (EOF) is reached. It is efficient when a large number of the records on the file need to be accessed at any given time. Data stored on a tape (sequential access) can be accessed only sequentially

Direct (Random) Access

Records are located by knowing their physical locations or addresses on the device rather than their positions relative to other records. Data stored on a CD device (direct-access) can be accessed either sequentially or randomly.

Types of Files used in an Organization System

Following are the types of files used in an organization system:

Master file: It contains the current information for a system. For example, customer file, student file, telephone directory.

Table file: It is a type of master file that changes infrequently and stored in a tabular format. For example, storing Zipcode.

Transaction file: It contains the day-to-day information generated from business activities. It is used to update or process the master file. For example, Addresses of the employees.

Temporary file: It is created and used whenever needed by a system.

Mirror files: They are the exact duplicates of other files. Help minimize the risk of downtime in cases when the original becomes unusable. They must be modified each time the original file is changed.

Log files: They contain copies of master and transaction records in order to chronicle any changes that are made to the master file. It facilitates auditing and provides mechanism for recovery in case of system failure.

Archive file: Backup files that contain historical versions of other files.

Documentation Control

Documentation is a process of recording the information for any reference or operational purpose. It helps users, managers, and IT staff, who require it. It is important that prepared document must be updated on regular basis to trace the progress of the system easily.

After the implementation of system if the system is working improperly, then documentation helps the administrator to understand the flow of data in the system to correct the flaws and get the system working.

Programmers or systems analysts usually create program and system documentation. Systems analysts usually are responsible for preparing documentation to help users learn the system. In large companies, a technical support team that includes technical writers might assist in the preparation of user documentation and training materials.

Advantages

1. It can reduce system downtime, cut costs, and speed up maintenance tasks.
2. It provides the clear description of formal flow of present system and helps to understand the type of input data and how the output can be produced.
3. It provides effective and efficient way of communication between technical and nontechnical users about system.
4. It facilitates the training of new user so that he can easily understand the flow of system.
5. It helps the user to solve the problems such as troubleshooting and helps the manager to take better final decisions of the organization system.
6. It provides better control to the internal or external working of the system.

Types of Documentations

When it comes to System Design, there are following four main documentations:

- Program documentation
- System documentation
- Operations documentation
- User documentation

Program Documentation

- It describes inputs, outputs, and processing logic for all the program modules.
- The program documentation process starts in the system analysis phase and continues during implementation.

- This documentation guides programmers, who construct modules that are well supported by internal and external comments and descriptions that can be understood and maintained easily.

Operations Documentation

Operations documentation contains all the information needed for processing and distributing online and printed output. Operations documentation should be clear, concise, and available online if possible.

It includes the following information:

- Program, systems analyst, programmer, and system identification.
- Scheduling information for printed output, such as report, execution frequency, and deadlines.
- Input files, their source, output files, and their destinations.
- E-mail and report distribution lists.
- Special forms required, including online forms.
- Error and informational messages to operators and restart procedures.
- Special instructions, such as security requirements.

User Documentation

It includes instructions and information to the users who will interact with the system. For example, user manuals, help guides, and tutorials. User documentation is valuable in training users and for reference purpose. It must be clear, understandable, and readily accessible to users at all levels.

The users, system owners, analysts, and programmers, all put combined efforts to develop a user's guide.

A user documentation should include:

- A system overview that clearly describes all major system features, capabilities, and limitations.
- Description of source document content, preparation, processing, and, samples.
- Overview of menu and data entry screen options, contents, and processing instructions.
- Examples of reports that are produced regularly or available at the user's request, including samples.
- Security and audit trail information.

- Explanation of responsibility for specific input, output, or processing requirements.
- Procedures for requesting changes and reporting problems.
- Examples of exceptions and error situations.
- Frequently asked questions (FAQs).
- Explanation of how to get help and procedures for updating the user manual.

System Documentation

System documentation serves as the technical specifications for the IS and how the objectives of the IS are accomplished. Users, managers and IS owners need never reference system documentation. System documentation provides the basis for understanding the technical aspects of the IS when modifications are made.

- It describes each program within the IS and the entire IS itself.
- It describes the system's functions, the way they are implemented, each
- program's purpose within the entire IS with respect to the order of execution, information passed to and from programs, and overall system flow.
- It includes data dictionary entries, data flow diagrams, object models, screen layouts, source documents, and the systems request that initiated the project
- Most of the system documentation is prepared during the system analysis and system design phases.
- During systems implementation, an analyst must review system documentation to verify that it is complete, accurate, and up-to-date, and including any changes made during the implementation process.