FACULTY NAME:J.VIJAY SATHIARAJ
DESIGNAION:GUEST LECTURER
DEPARTMENT:GEOGRAPHY
CLASS:M.TECH
SEMESTER:II

SUBJECT: Spatial Database Management

SUBJECT CODE: 22CC12

Unit-I: Introduction: An overview of database management system, Database system Vs file system, Database system concept and architecture, Data Definitions Language, Definition of GIS, DML, Database Structure, Entity Relationship Model – ER model concepts, Notation for ER diagram, Additional Features of the ER Model.

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.

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.

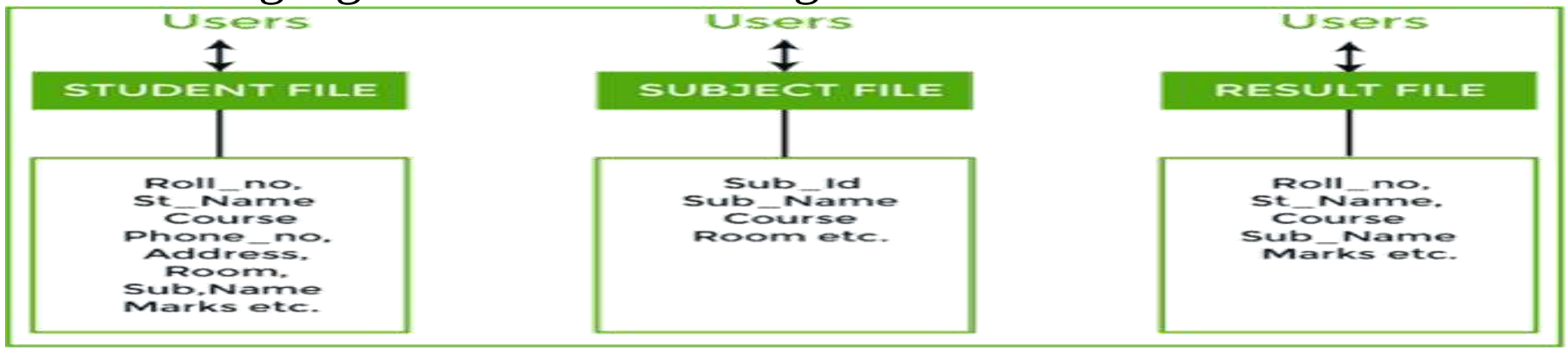
File Based Management System:

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



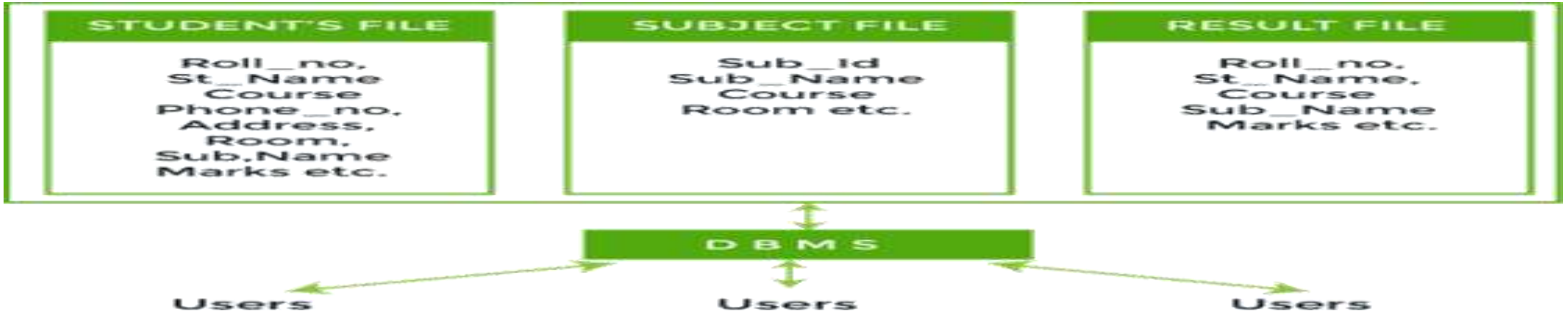
In the above figure:

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.



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.

Database Architectures

- Database architecture focuses on the design, development, implementation and maintenance of computer programs that store and organize information for businesses, agencies and institutions.
- A database architect develops and implements software to meet the needs of users.
- The design of a DBMS depends on its architecture. It can be centralized or decentralized or hierarchical.
- The architecture of a DBMS can be seen as either single tier or multi-tier. The tiers are classified as follows :
  - 1-tier architecture
  - 2-tier architecture
  - 3-tier architecture

- **1-tier architecture:**

- One-tier architecture involves putting all of the required components for a software application or technology on a single server or platform.
- One-tier architecture keeps all of the elements of an application, including the interface, Middleware and back-end data, in one place.
- Developers see these types of systems as the simplest and most direct way.

2 -tier architecture:

- The two-tier architecture is like client server application.
- The direct communication takes place between client and server. There is no intermediate between client and server.

3-tier architecture:

•The 3-Tier architecture contains another layer between the client and server. In this architecture, client can't directly communicate with the server.

•The application on the client-end interacts with an application server which further communicates with the database system.

•End user has no idea about the existence of the database beyond the application server. The database also has no idea about any other user beyond the application.

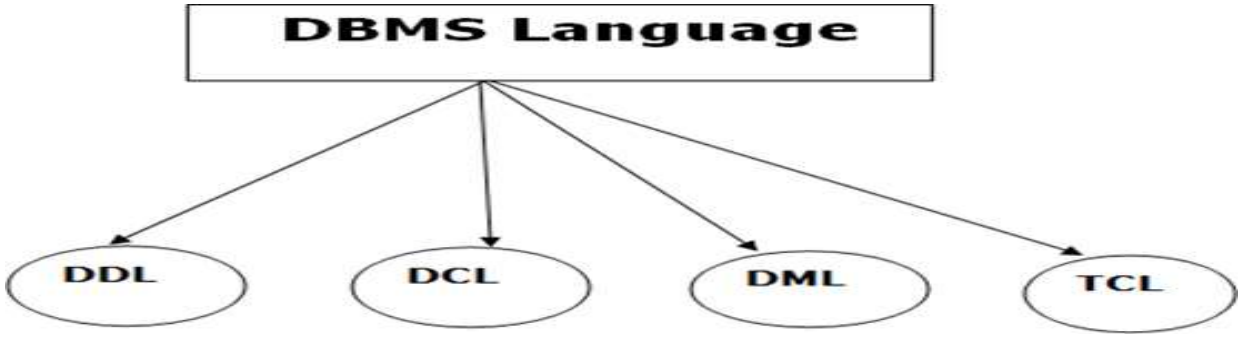
•It is the most widely used architecture to design a DBMS.

- It can be used in web applications and distributed applications

- **Database (Data) Tier:** Here, it contains the actual database along with its query processing languages. It also contains relations that define the data and their constraints.
- **Application (Middle) Tier:** It consists of the application server and the programs that access the database.
  - For a user, this application tier presents an abstracted view of the database. End-users are unaware of any existence of the database beyond the application.
  - At the other end, the database tier is not aware of any other user beyond the application tier. Hence, the application layer sits in the middle and acts as a mediator between the end-user and the database.
- **User (Presentation) Tier:**
  - End-users operate on this tier and they know nothing about any existence of the database beyond this layer.
  - At this layer, multiple views of the database can be provided by the application.
  - All views are generated by applications that reside in the application tier.

Database Language

- A DBMS has appropriate languages and interfaces to express database queries and updates.
- Database languages can be used to read, store and update the data in the database.



1. **Data Definition Language**

•DDL stands for Data Definition Language. It is used to define database structure or pattern.
•It is used to create schema, tables, indexes, constraints, etc. in the database.
•Using the DDL statements, you can create the skeleton of the database.
•Data definition language is used to store the information of metadata like the number of tables and schemas, their names, indexes, columns in each table, constraints, etc.
Here are some tasks that come under DDL:

- **Create:** It is used to create objects in the database.
- **Alter:** It is used to alter the structure of the database.
- **Drop:** It is used to delete objects from the database.
- **Truncate:** It is used to remove all records from a table.
- **Rename:** It is used to rename an object.
- **Comment:** It is used to comment on the data dictionary.

These commands are used to update the database schema that's why they come under Data definition language.

2.Data Manipulation Language

DML stands for Data Manipulation Language. It is used for accessing and manipulating data in a database. It handles user requests.

Here are some tasks that come under DML:

- **Select:** It is used to retrieve data from a database.
- **Insert:** It is used to insert data into a table.
- **Update:** It is used to update existing data within a table.
- **Delete:** It is used to delete all records from a table.
- **Merge:** It performs UPSERT operation, i.e., insert or update operations.

- **Call:** It is used to call a structured query language or a Java subprogram.
- **Explain Plan:** It has the parameter of explaining data.
- **Lock Table:** It controls concurrency

3.Data Control Language

•DCL stands for Data Control Language. It is used to retrieve the stored or saved data.

•The DCL execution is transactional. It also has rollback parameters.

(But in Oracle database, the execution of data control language does not have the feature of rolling back.)

Here are some tasks that come under DCL:

- **Grant:** It is used to give user access privileges to a database.
- **Revoke:** It is used to take back permissions from the user.

There are the following operations which have the authorization of Revoke: CONNECT, INSERT, USAGE, EXECUTE, DELETE, UPDATE and SELECT.

4.Transaction Control Language

TCL is used to run the changes made by the DML statement. TCL can be grouped into a logical transaction.

Here are some tasks that come under TCL:

- **Commit:** It is used to save the transaction on the database.
- **Rollback:** It is used to restore the database to original since the last Commit

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.

Facts about ER Diagram Model:

ER model allows you to draw Database Design

It is an easy to use graphical tool for modeling data Widely used in Database Design

It is a GUI representation of the logical structure of a Database

It helps you to identifies the entities which exist in a system and the relationships between those entities

Why use ER Diagrams?

Here, are prime reasons for using the ER Diagram

Helps you to define terms related to entity relationship modeling

Provide a preview of how all your tables should connect, what fields are going to be on each table

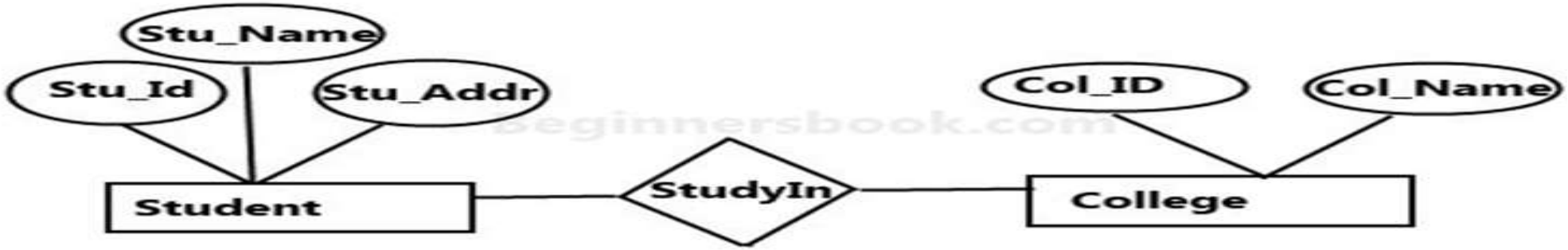
Helps to describe entities, attributes, relationships

ER diagrams are translatable into relational tables which allows you to build databases quickly ER diagrams can be used by database designers as a blueprint for implementing data in specific software applications

.

A simple ER Diagram:

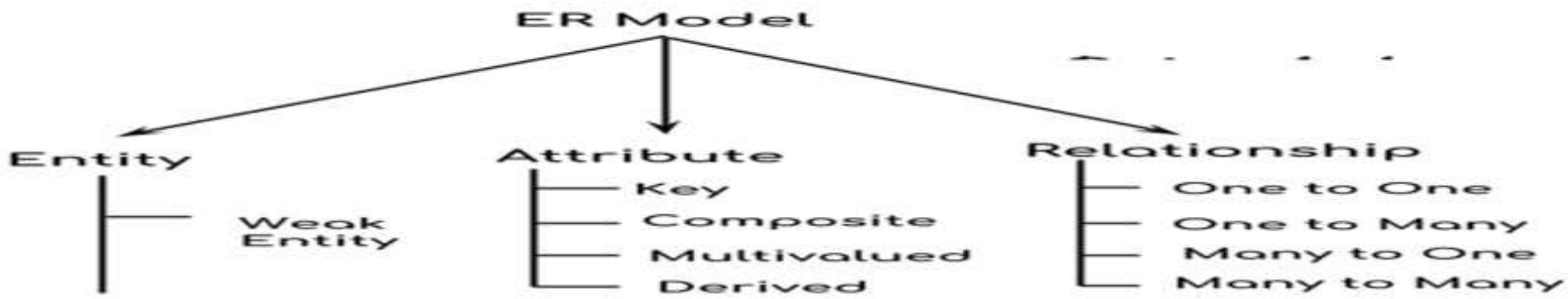
In the following diagram we have two entities Student and College and their relationship. The relationship between Student and College is many to one as a college can have many students however a student cannot study in multiple colleges at the same time. Student entity has attributes such as Stu_Id,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

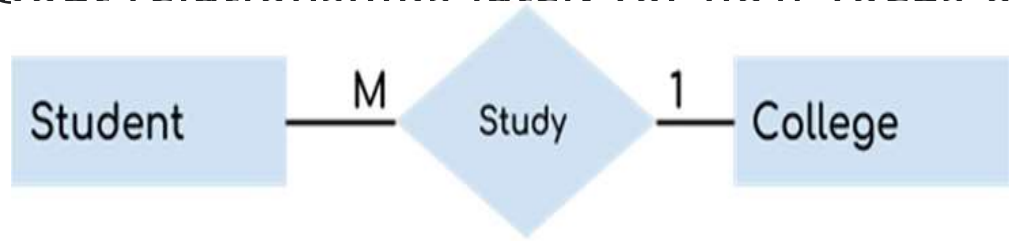
As shown in the above diagram, an ER diagram has three main components:

1. Entity
2. Attribute
3. Relationship

Entity

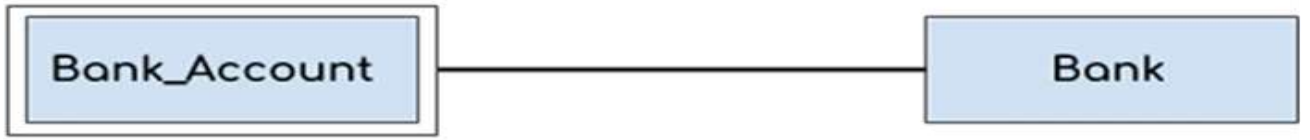
An entity is an object or component of data. An entity is represented as rectangle in an ER diagram.

For example: In the following ER diagram we have two entities Student and College and these two entities have many to one relationship as many students study in a single college. We will read more about relationships later, for now focus on entities.



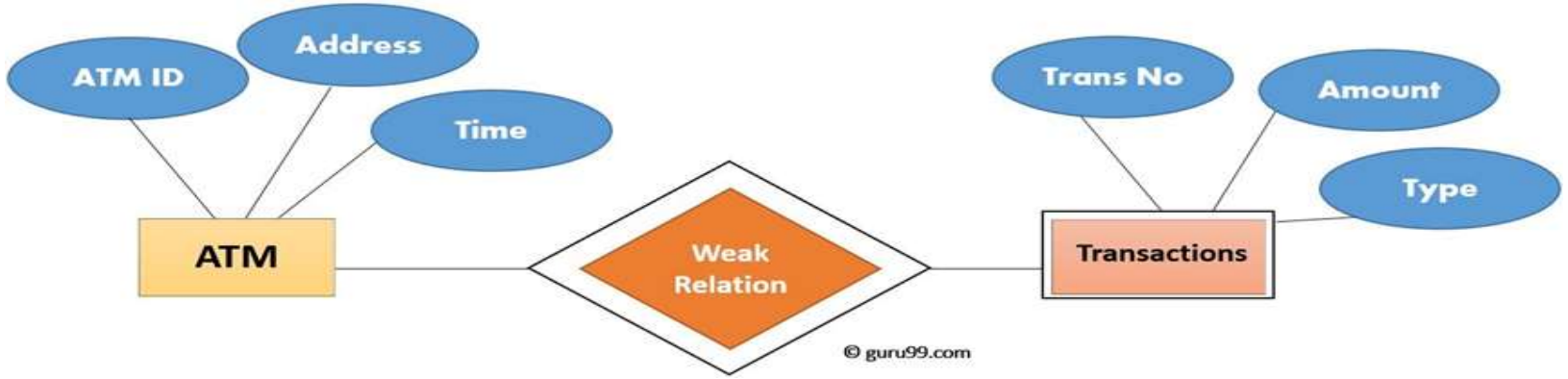
Weak Entity:

An entity that cannot be uniquely identified by its own attributes and relies on the relationship with other entity is called weak entity. The weak entity is represented by a double rectangle. For example – a bank account cannot be uniquely identified without knowing the bank to which the account belongs, so bank account is a weak entity.



Weak Entities

A weak entity is a type of entity which doesn't have its key attribute. It can be identified uniquely by considering the primary key of another entity. For that, weak entity sets need to have participation.



In above example, "Trans No" is a discriminator within a group of transactions in an ATM.Let's learn more about a weak entity by comparing it with a Strong Entity

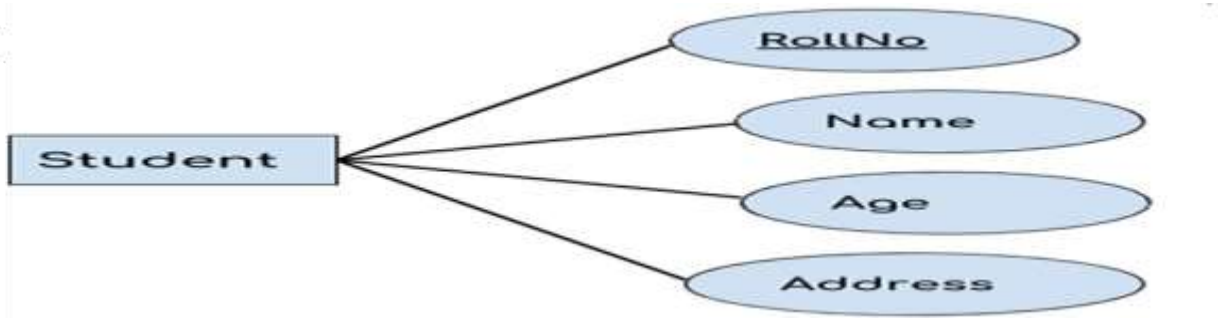
1. **Attribute**

An attribute describes the property of an entity. An attribute is represented as Oval in an ER diagram. There are four types of attributes:

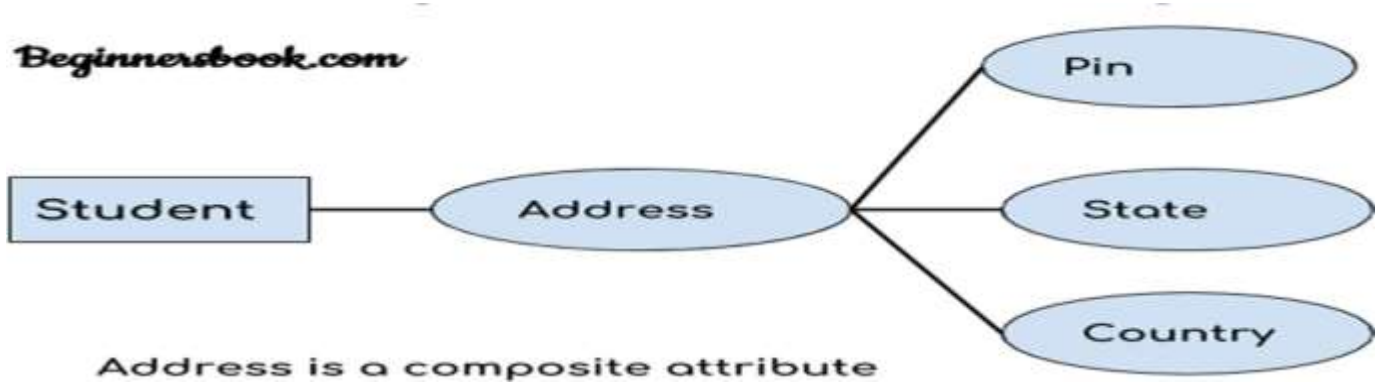
1. Key attribute
2. Composite attribute
3. Multivalued attribute
4. Derived attribute

Key attribute:

A key attribute can uniquely identify an entity from an entity set. For example, student roll number can uniquely identify a student from a set of students. Key attribute is represented by oval same as other attributes however the **text of key attribute is underlined**.



•**Composite attribute:** An attribute that is a combination of other attributes is known as composite attribute. For example, In student entity, the student address is a composite attribute as an address is composed of other attributes such as pin code, state, country.

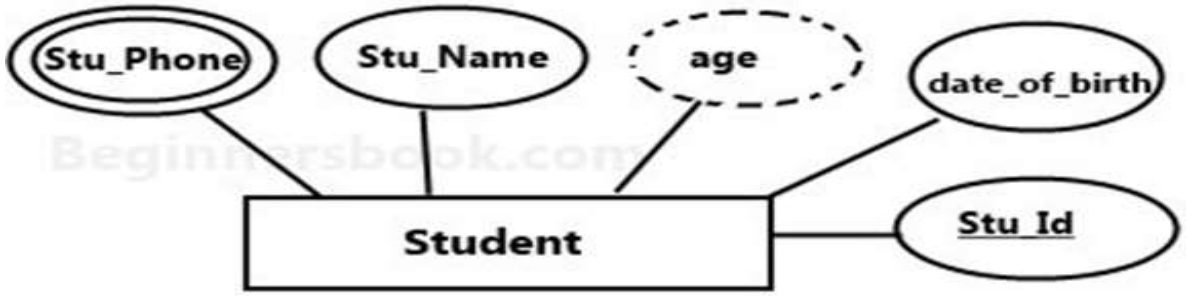


1. **Multivalued attribute:**

An attribute that can hold multiple values is known as multivalued attribute. It is represented with **double ovals** in an ER Diagram. For example – A person can have more than one phone numbers so the phone number attribute is multivalued.

Derived attribute:

A derived attribute is one whose value is dynamic and derived from another attribute. It is represented by **dashed oval** in an ER Diagram. For example – Person age is a derived attribute as it changes over time and can be derived from another attribute (Date of birth).



•**Relationship**

Cardinality: Defines the numerical attributes of the relationship between two entities or entity sets.

A relationship is represented by diamond shape in ER diagram, it shows the relationship among entities. There are four types of cardinal relationships:

•One to One

•One to Many

•Many to One

•Many to Many

•**One to One Relationship**

•When a single instance of an entity is associated with a single instance of another entity then it is called one to one relationship. For example, a person has only one passport and a passport is given to one person.

1. **One to Many Relationship**

When a single instance of an entity is associated with more than one instances of another entity then it is called one to many relationship. For example – a customer can place many orders but a order cannot be placed by many customers.

- **Many to One Relationship**
- When more than one instances of an entity is associated with a single instance of another entity then it is called many to one relationship. For example – many students can study in a single college but a student cannot study in many colleges at the same time.

1. **Many to Many Relationship**

When more than one instances of an entity is associated with more than one instances of another entity then it is called many to many relationship. For example, a can be assigned to many projects and a project can be assigned to many students.

Advantages of ER Model

Simple: Conceptually ER Model is very easy to build. If we know the relationship between the attributes and the entities we can easily build the ER Diagram for the model.

- ***Effective Communication Tool***: This model is used widely by the database designers for communicating their ideas.
- ***Easy Conversion to any Model***: This model maps well to the relational model and can be easily converted relational model by converting the ER model to the table. This model can also be converted to any other model like network model, hierarchical model etc.

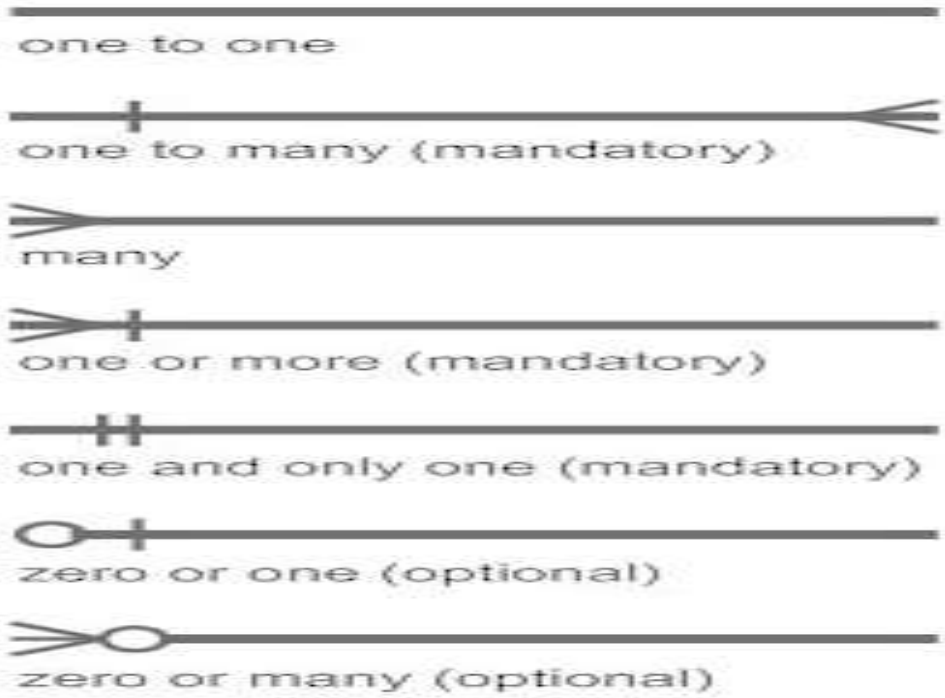
Disadvantages of ER Model

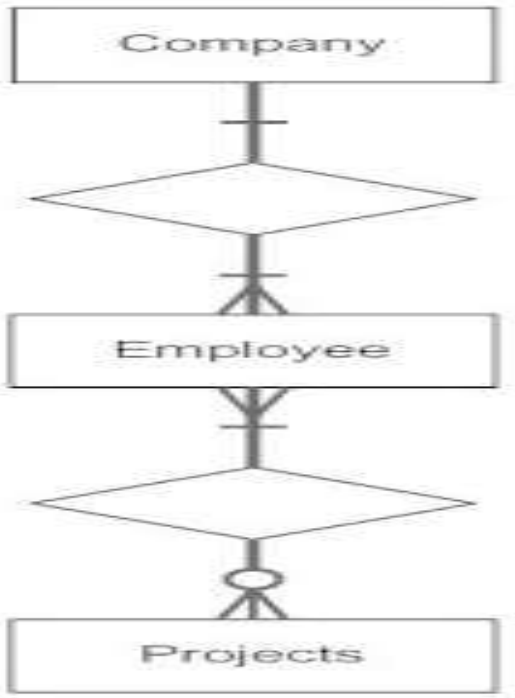
No industry standard for notation: There is no industry standard for developing an ER model. So one developer might use notations which are not understood by other developers.

- ***Hidden information:*** Some information might be lost or hidden in the ER model. As it is a high-level view so there are chances that some details of information might be hidden.

Notation of ER diagram

Database can be represented using the notations. In ER diagram, many notations are used to express the cardinality. These notations are as follows:





Additional Features of the ER Model in DBMS

As the complexity of data increased, it became more and more difficult to use the traditional ER Model for database modelling. Hence some Additional Features were made to the existing ER Model to make it able to handle the complex applications better.

Hence, Three new concepts were added to the existing ER Model, they were:

Generalization

Specialization

Aggregration

Some entities have relationships that form hierarchies. For instance, Employee can be an hourly employee or contracted employee.

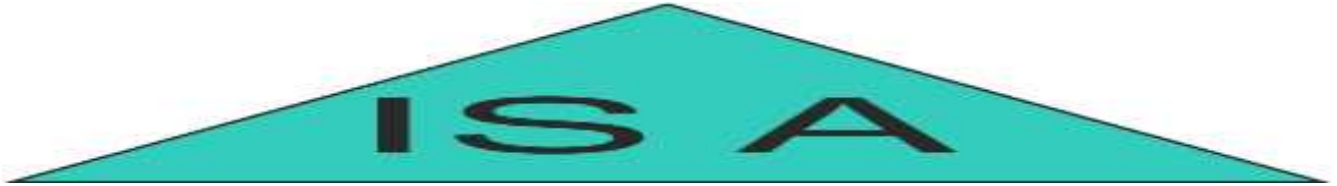
In this relationship hierarchies, some entities can act as superclass and some other entities can act as subclass.

Superclass: An entity type that represents a general concept at a high level, is called superclass.

Subclass: An entity type that represents a specific concept at lower levels, is called subclass.

The subclass is said to inherit from superclass. When a subclass inherits from one or more super classes, it inherits all their attributes.

The symbol used for specialization/ Generalization is



Generalization

Generalization is a process of extracting common properties from a set of entities and creating a generalized entity from it. It is a bottom-up approach, and it helps to reduce the size and complexity of the schema.

Example: Let us take two low-level entities as Car and Bus, and these two will have many common attributes and some specific attributes. And We will generalize and link the common attributes to the newly formed high-level entity named Vehicle.

Specialization

Specialization is opposite to Generalization. In this, entity is divided into subentities bases on their charactertics(distinguishing features). It breaks an entity into multiple entities from higher level to lower level. It is a top down approach.

Aggregation

Aggregation refers to the process by which entities are combined to form a single meaningful entity. The specific entities are combined because they do not make sense on their own. To establish a single entity, aggregation creates a relationship that combines these entities. The resulting entity makes sense because it enables the system to function well.