DBMS COMPLETE UNIT - 1

For better understanding watch this youtube playlist along with the topics covered in this written material--

https://youtube.com/playlist?list=PLxCzCOWd7aiFAN6I8CuViBuCdJgiOkT2Y

What is Data?

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' originated from the word 'datum' that means 'single piece of information.' It is the 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.

What is Database?

A **database** is an organized collection of data, so that it can be easily accessed and managed.

You can organize data into tables, rows, columns, and index it to make it easier to find relevant information.

Database handlers create a database in such a way that only one set of software programs provides access of data to all the users.

The **main purpose** of the database is to operate a large amount of information by storing, retrieving, and managing data.

There are many **dynamic websites** on the World Wide Web nowadays which are handled through databases. For example, a model that checks the availability of rooms in a hotel. It is an example of a dynamic website that uses a database.

There are many **databases available** like MySQL, Sybase, Oracle, MongoDB, Informix, PostgreSQL, SQL Server, etc.

Modern databases are managed by the database management system (DBMS).

https://cgccollegespace.live

SQL or Structured Query Language is used to operate on the data stored in a database. SQL depends on relational algebra and tuple relational calculus.

A cylindrical structure is used to display the image of a database.

DBMS (Database Management System)

Database management System is software which is used to store and retrieve the database. For example, Oracle, MySQL, etc.; these are some popular DBMS tools.

- DBMS provides the interface to perform the various operations like creation, deletion, modification, etc.
- DBMS allows the user to create their databases as per their requirement.
- DBMS accepts the request from the application and provides specific data through the operating system.
- DBMS contains the group of programs which acts according to the user instruction.
- It provides security to the database.

Advantage of DBMS

Controls redundancy

It stores all the data in a single database file, so it can control data redundancy.

Data sharing

An authorized user can share the data among multiple users.

Backup

It providesBackup and recovery subsystem. This recovery system creates automatic data from system failure and restores data if required.

Multiple user interfaces

It provides a different type of user interfaces like GUI, application interfaces.

Disadvantage of DBMS

Size

It occupies large disk space and large memory to run efficiently.

Cost

DBMS requires a high-speed data processor and larger memory to run DBMS software, so it is costly.

Complexity

DBMS creates additional complexity and requirements.

RDBMS (Relational Database Management System)

The word RDBMS is termed as 'Relational Database Management System.' It is represented as a table that contains rows and columns.

RDBMS is based on the Relational model; it was introduced by E. F. Codd.

A relational database contains the following components:

- Table
- Record/ Tuple
- Field/Column name /Attribute
- Instance
- Schema
- Keys

An RDBMS is a tabular DBMS that maintains the security, integrity, accuracy, and consistency of the data.

DBMS vs. File System

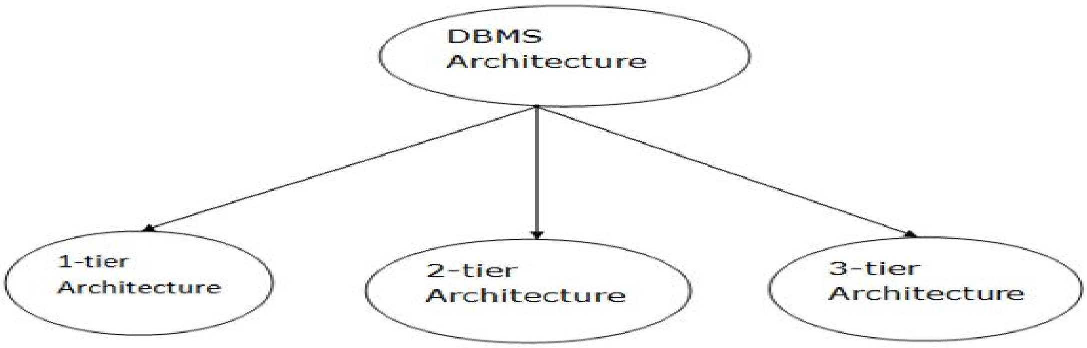
There are following differences between DBMS and File system:

DBMS	File System
DBMS is a collection of data. In DBMS, the user is not required to write the procedures.	File system is a collection of data. In this system, the user has to write the procedures for managing the database.
DBMS gives an abstract view of data that hides the details.	File system provides the detail of the data representation and storage of data.
DBMS provides a crash recovery mechanism, i.e., DBMS protects the user from system failure.	File system doesn't have a crash mechanism, i.e., if the system crashes while entering some data, then the content of the file will be lost.
DBMS provides a good protection mechanism.	It is very difficult to protect a file under the file system.
DBMS contains a wide variety of sophisticated techniques to store and retrieve the data.	File system can't efficiently store and retrieve the data.
DBMS takes care of Concurrent access of data using some form of locking.	In the File system, concurrent access has many problems like redirecting the file while deleting some information or updating some information.

DBMS Architecture

- The DBMS design depends upon its architecture. The basic client/server architecture is used to deal with a large number of PCs, web servers, database servers and other components that are connected with networks.
- The client/server architecture consists of many PCs and a workstation which are connected via the network.
- DBMS architecture depends upon how users are connected to the database to get their request done.

Types of DBMS Architecture



Database architecture can be seen as a single tier or multi-tier. But logically, database architecture is of two types like: **2-tier architecture** and **3-tier architecture**.

1-Tier Architecture

- In this architecture, the database is directly available to the user. It means the user can directly sit on the DBMS and uses it.
- Any changes done here will directly be done on the database itself. It doesn't provide a handy tool for end users.

- The 1-Tier architecture is used for development of the local application, where programmers can directly communicate with the database for the quick response.

2-Tier Architecture

- The 2-Tier architecture is the same as basic client-server. In the two-tier architecture, applications on the client end can directly communicate with the database at the server side. For this interaction, API's like: **ODBC**, **JDBC** are used.
- The user interfaces and application programs are run on the client-side.
- The server side is responsible to provide the functionalities like: query processing and transaction management.
- To communicate with the DBMS, client-side application establishes a connection with the server side.

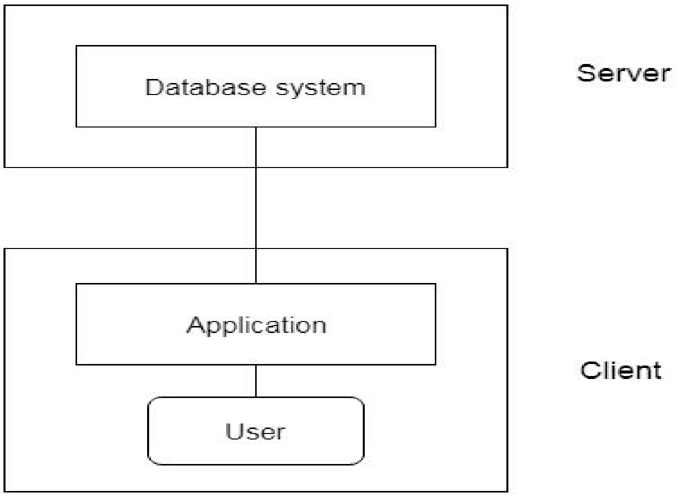


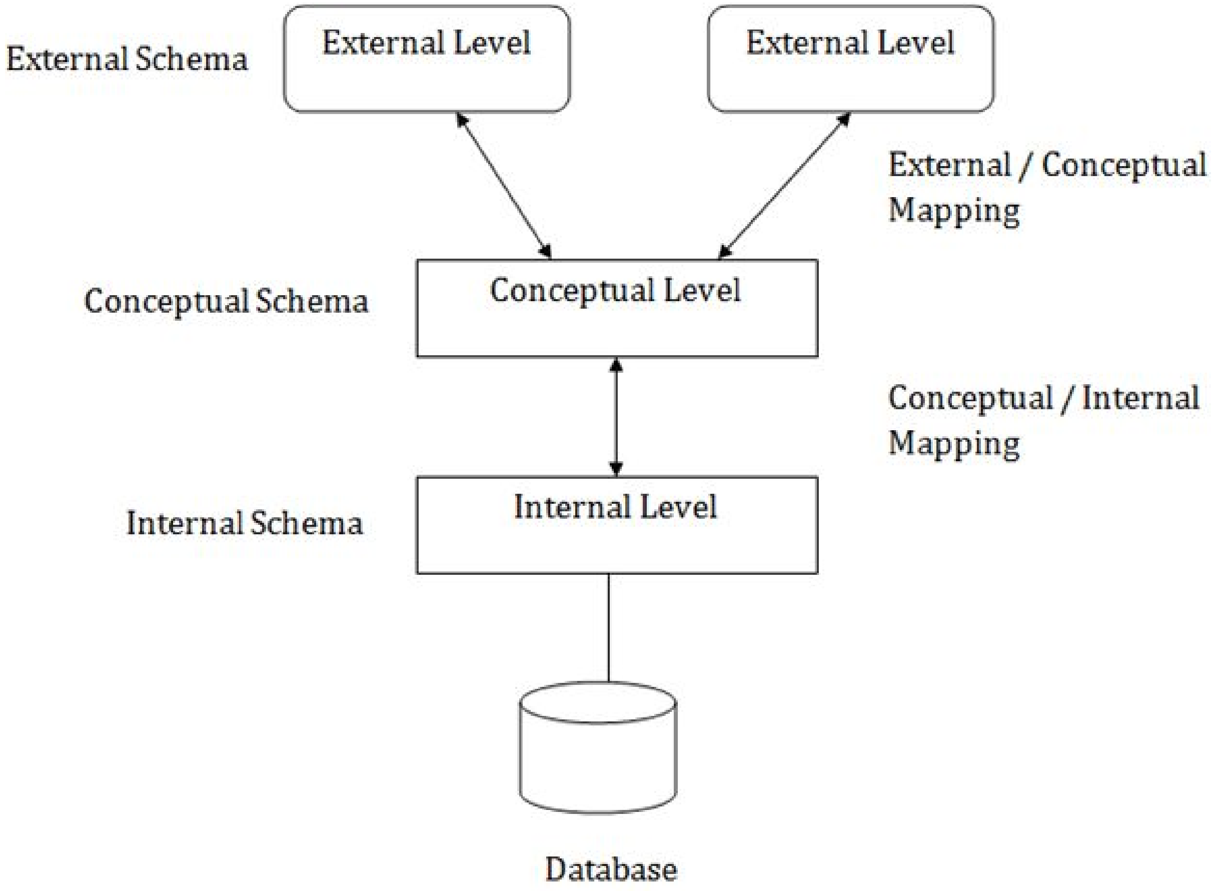
Fig: 2-tier Architecture

3-Three schema Architecture

- The three schema architecture is also called ANSI/SPARC architecture or three-level architecture.

- This framework is used to describe the structure of a specific database system.
- The three schema architecture is also used to separate the user applications and physical database.
- The three schema architecture contains three-levels. It breaks the database down into three different categories.

The three-schema architecture is as follows:



In the above diagram:

- It shows the DBMS architecture.
- Mapping is used to transform the request and response between various database levels of architecture.
- Mapping is not good for small DBMS because it takes more time.

- In External / Conceptual mapping, it is necessary to transform the request from external level to conceptual schema.
- In Conceptual / Internal mapping, DBMS transform the request from the conceptual to internal level.

1. Internal Level

- The internal level has an internal schema which describes the physical storage structure of the database.
- The internal schema is also known as a physical schema.
- It uses the physical data model. It is used to define that how the data will be stored in a block.
- The physical level is used to describe complex low-level data structures in detail.

2. Conceptual Level

- The conceptual schema describes the design of a database at the conceptual level. Conceptual level is also known as logical level.
- The conceptual schema describes the structure of the whole database.
- The conceptual level describes what data are to be stored in the database and also describes what relationship exists among those data.
- In the conceptual level, internal details such as an implementation of the data structure are hidden.
- Programmers and database administrators work at this level.

3. External Level

- At the external level, a database contains several schemas that sometimes called as subschema. The subschema is used to describe the different view of the database.
- An external schema is also known as view schema.
- Each view schema describes the database part that a particular user group is interested and hides the remaining database from that user group.
- The view schema describes the end user interaction with database systems.

Data Abstraction

Database systems comprise complex data-structures. In order to make the system efficient in terms of retrieval of data, and reduce complexity in terms of usability of users, developers use abstraction i.e. hide irrelevant details from the users. This approach simplifies database design.

There are mainly **3** levels of data abstraction:

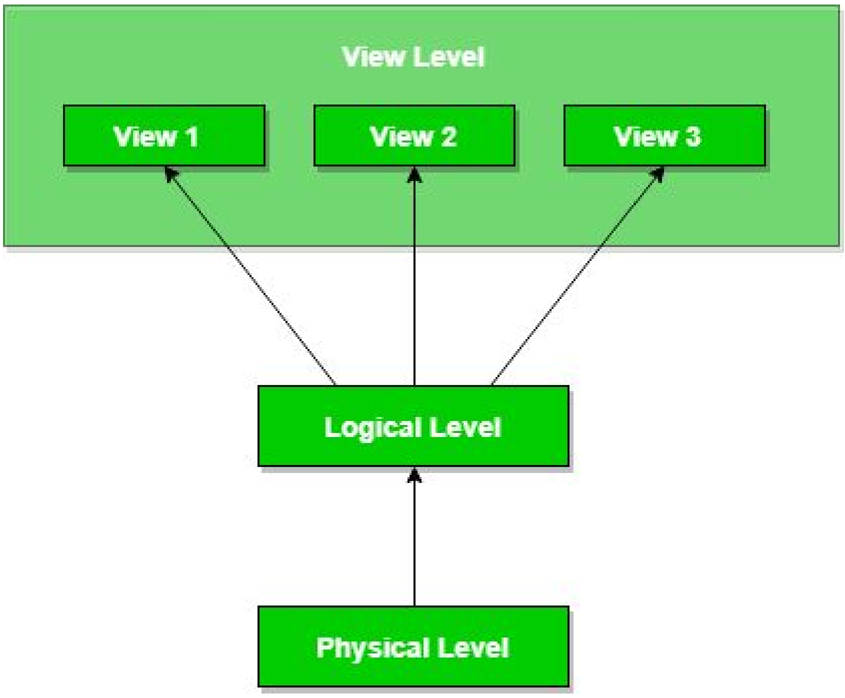
Physical: This is the lowest level of data abstraction. It tells us how the data is actually stored in memory. The access methods like sequential or random access and file organisation methods like B+ trees, hashing used for the same. Usability, size of memory, and the number of times the records are factors which we need to know while designing the database.

Suppose we need to store the details of an employee. Blocks of storage and the amount of memory used for these purposes is kept hidden from the user.

Logical: This level comprises the information that is actually stored in the database in the form of tables. It also stores the relationship among the data entities in relatively simple structures. At this level, the information available to the user at the view level is unknown.

We can store the various attributes of an employee and relationships, e.g. with the manager can also be stored.

View: This is the highest level of abstraction. Only a part of the actual database is viewed by the users. This level exists to ease the accessibility of the database by an individual user. Users view data in the form of rows and columns. Tables and relations are used to store data. Multiple views of the same database may exist. Users can just view the data and interact with the database, storage and implementation details are hidden from them.



The main purpose of data abstraction is achieving data independence in order to save time and cost required when the database is modified or altered.

Data Independence

- Data independence can be explained using the three-schema architecture.
- Data independence refers to the characteristic of being able to modify the schema at one level of the database system without altering the schema at the next higher level.

There are two types of data independence:

1. Logical Data Independence

- Logical data independence refers to the characteristic of being able to change the conceptual schema without having to change the external schema.It refers characteristic of being able to modify the logical schema without affecting the external schema or application program
- These changes may include insertion or deletion of attributes, altering table structures entities or relationships to the logical schema etc.
- Logical data independence is used to separate the external level from the conceptual view.
- If we do any changes in the conceptual view of the data, then the user view of the data would not be affected.
- Logical data independence occurs at the user interface level.

2. Physical Data Independence

- Physical data independence can be defined as the capacity to change the internal schema without having to change the conceptual schema. It refers to the characteristic of being able to modify the physical schema without any alterations to the conceptual or logical schema

- e.g., Conceptual structure of the database would not be affected by any change in storage size of the database system server.
- If we do any changes in the storage size of the database system server, then the Conceptual structure of the database will not be affected.
- Physical data independence is used to separate conceptual levels from the internal levels.
- Physical data independence occurs at the logical interface level.

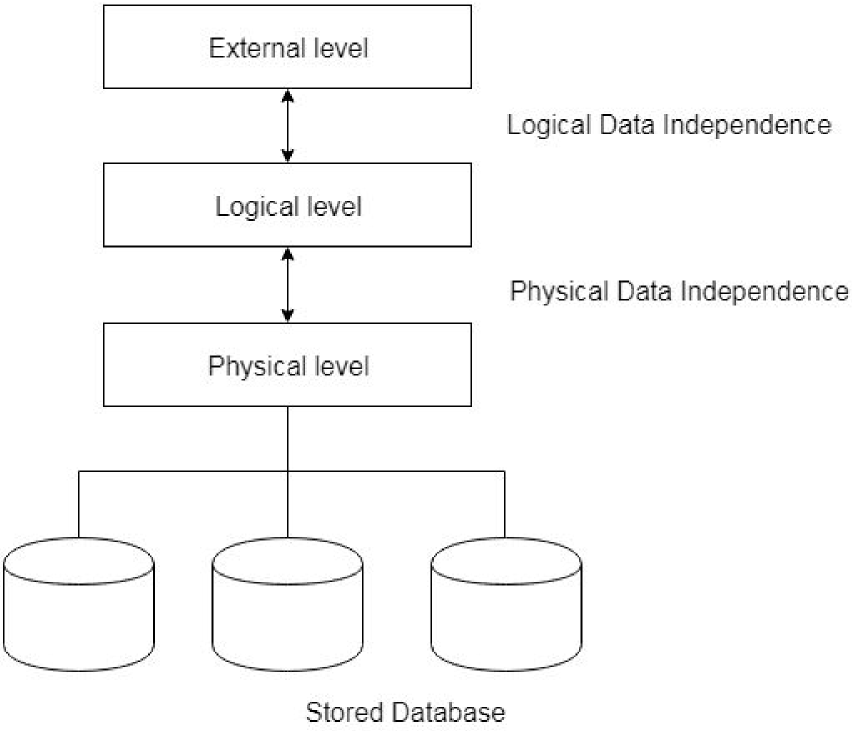
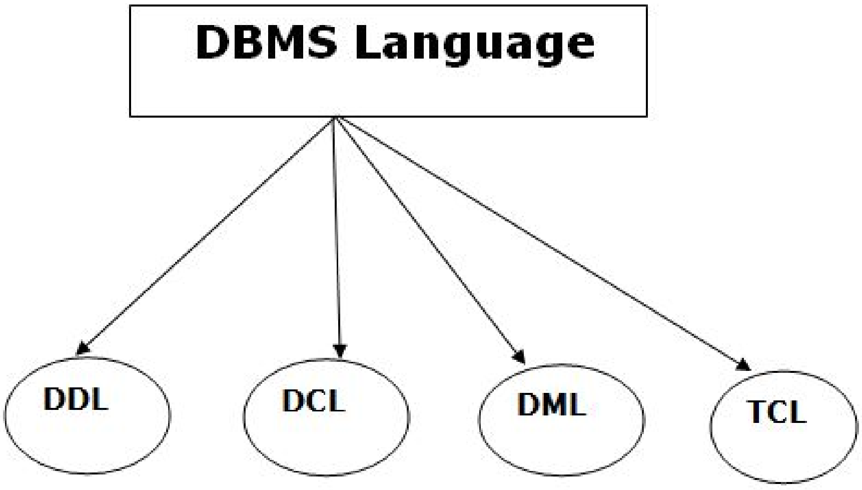


Fig: Data Independence

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.

Types of Database Language



1. Data Definition Language

- **DDL** stands for **D**ata **D**efinition **L**anguage. 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.

  For more details https://www.geeksforgeeks.org/sql-create/

- **Alter:** It is used to alter the structure of the database.

  For more details -- https://www.geeksforgeeks.org/sql-alter-add-drop-modify/

- **Drop:** It is used to delete objects from the database.

  For more details --https://www.geeksforgeeks.org/sql-drop-truncate/

- **Truncate:** It is used to remove all records from a table.

  For more details -- https://www.geeksforgeeks.org/sql-drop-truncate/

- **Rename:** It is used to rename an object.

  For more details -- https://www.geeksforgeeks.org/sql-alter-rename/

- **Comment:** It is used to comment on the data dictionary.

  For more details -- https://www.geeksforgeeks.org/sql-comments/

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 **D**ata **M**anipulation **L**anguage. 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.(DQL - Data Query Language)

  For more details -- https://www.geeksforgeeks.org/sql-select-query/

- **Insert:** It is used to insert data into a table.

  For more details --https://www.geeksforgeeks.org/sql-insert-statement/

- **Update:** It is used to update existing data within a table.

  For more details --https://www.geeksforgeeks.org/sql-update-statement/

- **Delete:** It is used to delete all records from a table.

  For more details --https://www.geeksforgeeks.org/sql-delete-statement/

- **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 **D**ata **C**ontrol **L**anguage. 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.

For more details visit --

- https://www.geeksforgeeks.org/difference-between-grant-and-revoke/#:~:text=SQL%20Grant%20command%20is%20specifically,permissions%20for%20other%20users%20too.

or

- https://www.studytonight.com/dbms/dcl-command.php

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.
- **SAVEPOINT**–sets a savepoint within a transaction.
- **SET TRANSACTION**–specify characteristics for the transaction.

For more details --

https://www.geeksforgeeks.org/sql-transactions/

or

https://www.tutorialspoint.com/sql/sql-transactions.htm

Data Models

Data Model is the modeling 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. A database model shows the logical structure of a database, including the relationships and constraints that determine how data can be stored and accessed. Individual database models are designed based on the rules and concepts of whichever broader data model the designers adopt. Most data models can be represented by an accompanying database diagram.

Types of database models

There are many kinds of data models. Some of the most common ones include:

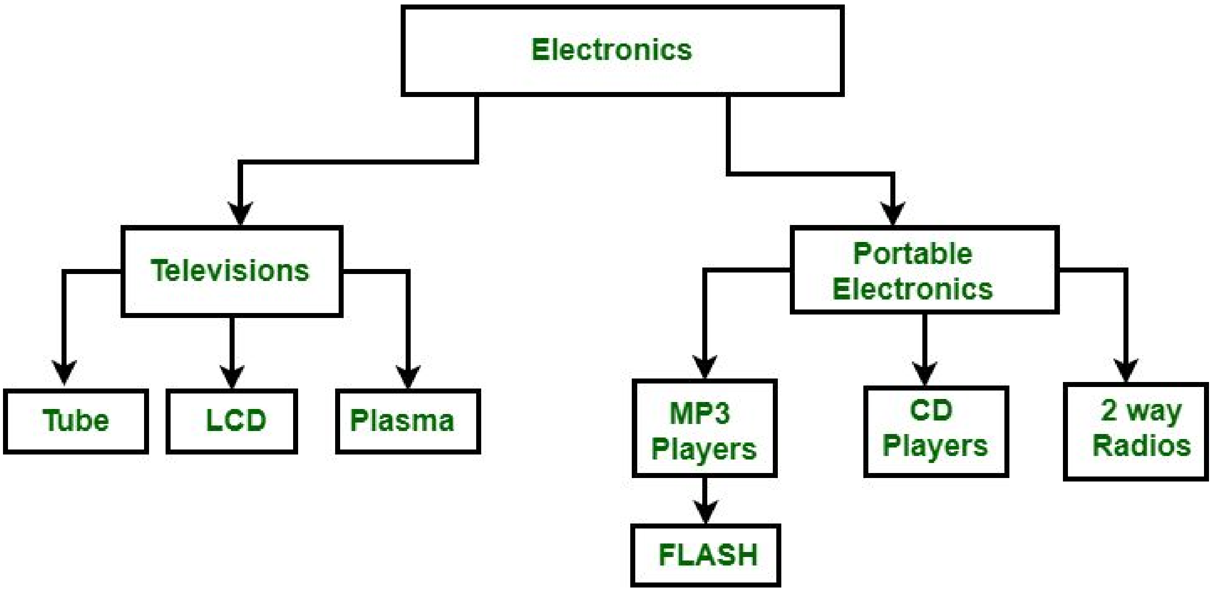
- Hierarchical database model
- Relational model
- Network model
- Object-oriented database model
- Entity-relationship model

Hierarchical model

The hierarchical model organizes data into a tree-like structure, where each record has a single parent or root. Sibling records are sorted in a particular order. That order is used as the physical order for storing the database. This model is good for describing many real-world relationships. This model was primarily used by IBM's Information Management Systems in the 60s and 70s, but they are rarely seen today due to certain operational inefficiencies.

Hierarchical data model is the oldest type of data model. It was developed by IBM in 1968. It organizes data in tree-like structure. Hierarchical model consists of following :

- It contains nodes which are connected by branches.
- Topmost node is called the root node.
- If multiple nodes appear at top level, then these can be called as root segments.
- Each node has exactly one parent.
- One parent may have many children.



In the above figure, Electronics is root node which has two children i.e. Televisions and Portable Electronics. These two have further children for which they act as parents. For example: Television has children as Tube, LCD and Plasma, for these three Television act as parents. It follows one to many relationships.

ADVANTAGES:

• Hierarchical Model is simple to construct and operate on depicts a set of one-to-many (1:M) relationships between a parent and its children segments

– Each parent can have many children

– each child has only one parent

DISADVANTAGES:

• Navigational and procedural nature of processing

• Database is visualized as a linear arrangement of records

• Little scope for "query optimization“

• Wastage of storage space

• Inconsistency during updation of database because when parent node is deleted that results in deletion of child node force fully.

• Not flexible.

https://cgccollegespace.live

Commercially available Hierarchical

Database system:

- IBM’s information management system
- MRI’s system 2000
- IMS informatics Mark IV
- Time –shared Data management System of SDC

Relational model

The most common model, the relational model sorts data into tables, also known as relations, each of which consists of columns and rows. Each column lists an attribute of the entity in question, such as price, zip code, or birth date. Together, the attributes in a relation are called a domain. A particular attribute or combination of attributes is chosen as a primary key that can be referred to in other tables, when it’s called a foreign key.

Each row, also called a tuple, includes data about a specific instance of the entity in question, such as a particular employee.

The model also accounts for the types of relationships between those tables, including one-to-one, one-to-many, and many-to-many relationships. Here’s an example:

https://d2slcw3kip6qmk.cloudfront.net/marketing/pages/chart/seo/database/discovery/relational-model.svg

Within the database, tables can be normalized, or brought to comply with normalization rules that make the database flexible, adaptable, and scalable. When normalized, each piece of data is atomic, or broken into the smallest useful pieces.

Relational databases are typically written in Structured Query Language (SQL). The model was introduced by E.F. Codd in 1970.

The Relational Model represents how data is stored in Relational Databases. A relational database stores data in the form of relations (tables). Consider a relation STUDENT with attributes ROLL_NO, NAME, ADDRESS, PHONE and AGE shown in Table 1.

STUDENT

ROLL_NO	NAME	ADDRESS	PHONE	AGE
1	RAM	DELHI	9455123451	18
2	RAMESH	GURGAON	9652431543	18
3	SUJIT	ROHTAK	9156253131	20
4	SURESH	DELHI		18

IMPORTANT TERMINOLOGIES

- **Attribute:** Attributes are the properties that define a relation. e.g.; **ROLL_NO**, **NAME**
- **Relation Schema:** A relation schema represents the name of the relation with its attributes. e.g.; STUDENT (ROLL_NO, NAME, ADDRESS, PHONE and AGE) is relation schema for STUDENT. If a schema has more than 1 relation, it is called Relational Schema.
- **Tuple:** Each row in the relation is known as tuple. The above relation contains 4 tuples, one of which is shown as:

1	RAM	DELHI	9455123451	18

- **Relation Instance:** The set of tuples of a relation at a particular instance of time is called a relation instance. Table 1 shows the relation instance of STUDENT at a particular time. It can change whenever there is insertion, deletion or updation in the database.
- **Degree:** The number of attributes in the relation is known as degree of the relation. The **STUDENT** relation defined above has degree 5.
- **Cardinality:** The number of tuples in a relation is known as cardinality. The **STUDENT** relation defined above has cardinality 4.
- **Column:** Column represents the set of values for a particular attribute. The column **ROLL_NO** is extracted from relation STUDENT.

ROLL_NO
1
2
3
4

- **NULL Values:** The value which is not known or unavailable is called NULL value. It is represented by blank space. e.g.; PHONE of STUDENT having ROLL_NO 4 is NULL.

ADVANTAGES:-

- **Ease of use:** The revision of any information as tables consisting of rows and columns is quite natural and therefore even first time users find it attractive.

2. Flexibility: Different tables from which information has to be linked and extracted can be easily manipulated by operators such as project and join to give information in the form in which it is desired.

3. Precision: The usage of relational algebra and relational calculus in the manipulation of the relations between the tables ensures that there is no ambiguity.

4. Security: Security control and authorization can also be implemented more easily by moving sensitive attributes in a given table into a separate relation with its own authorization controls.

5. Structural Independence

6. Easy to Design

Disadvantage:-

- A major constraint and therefore disadvantage in the use of relational database systems is machine performance. If the number of tables between which relationships to be established are large and the tables themselves are voluminous, the performance in responding to queries is definitely degraded.

- Need more powerful computing H/W and data storage devices that increase the cost and H/W overhead.

Network Data Model :

It is the advanced version of the hierarchical data model. To organize data it uses directed graphs instead of the tree-structure. In this child can have more than one parent. It uses the concept of the two data structures i.e. Records and Sets.

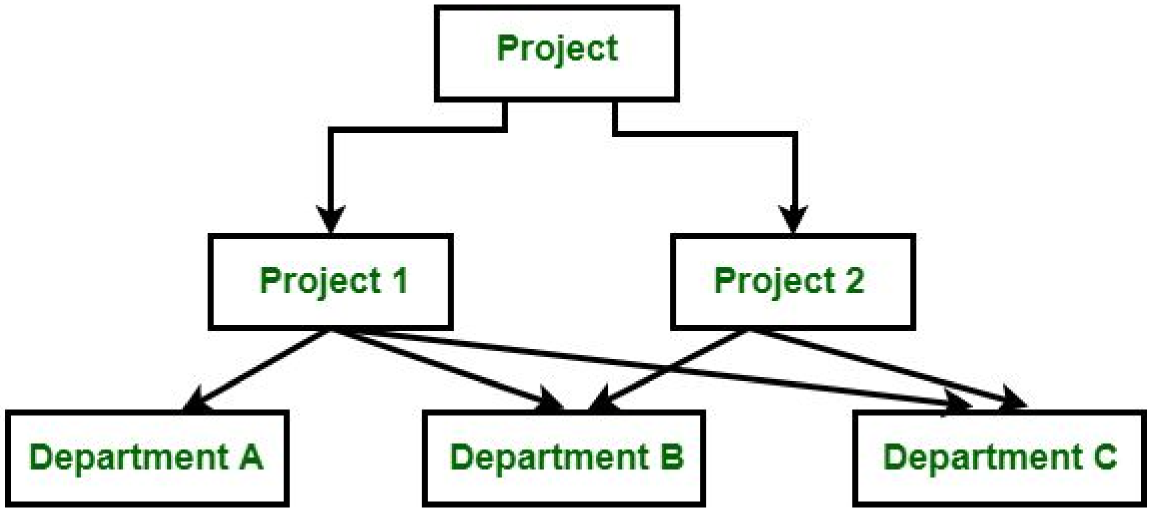


Figure – Network Data Model

In the above figure, Project is the root node which has two children i.e. Project 1 and Project 2. Project 1 has 3 children and Project 2 has 2 children. There are 5 children i.e Department A, Department B and Department C, they are network related children as we said that this model can have more than one parent. So, for the Department B and Department C have two parents i.e. Project 1 and Project 2.

ADVANTAGES

- Provide very efficient "High-speed" retrieval

- Simplicity : The network model is conceptually simple and easy to design.

- Ability to handle more relationship types: The network model can handle the one-to-many and many-to-many relationships.

- Ease of data access: In the network database terminology, a relationship is a set. Each set comprises of two types of records.- an owner record and a member record, In a network model an application can access an owner record and all the member records within a set..

- Data Independence: The network model draws a clear line of demarcation between programs and the complex physical storage details. The application programs work independently of the data. Any changes made in the data characteristics do not affect the application program.

- Flexible than hierarchical models.

DISADVANTAGES

- Complex to design than relational model

- Maintenance is not easy.

- Flexibility and efficiency are less than relational model

- Lack of Structural independence.: Making structural modifications to the database is very difficult in the network database model as the data access method is navigational. Any changes made to the database structure require the application programs to be modified before they can access data. Though the network model achieves data independence, it still fails to achieve structural independence.

Object-oriented database model

This model defines a database as a collection of objects, or reusable software elements, with associated features and methods. There are several kinds of object-oriented databases:

A **multimedia database** incorporates media, such as images, that could not be stored in a relational database.

A **hypertext database** allows any object to link to any other object. It's useful for organizing lots of disparate data, but it's not ideal for numerical analysis.

The object-oriented database model is the best known post-relational database model, since it incorporates tables, but isn't limited to tables. Such models are also known as hybrid database models.Object oriented data model is based upon real world situations. These situations are represented as objects, with different attributes. All these objects have multiple relationships between them.

Elements of Object oriented data model

Objects

The real world entities and situations are represented as objects in the Object oriented database model.

Attributes and Method

Every object has certain characteristics. These are represented using Attributes. The behaviour of the objects is represented using Methods.

Class

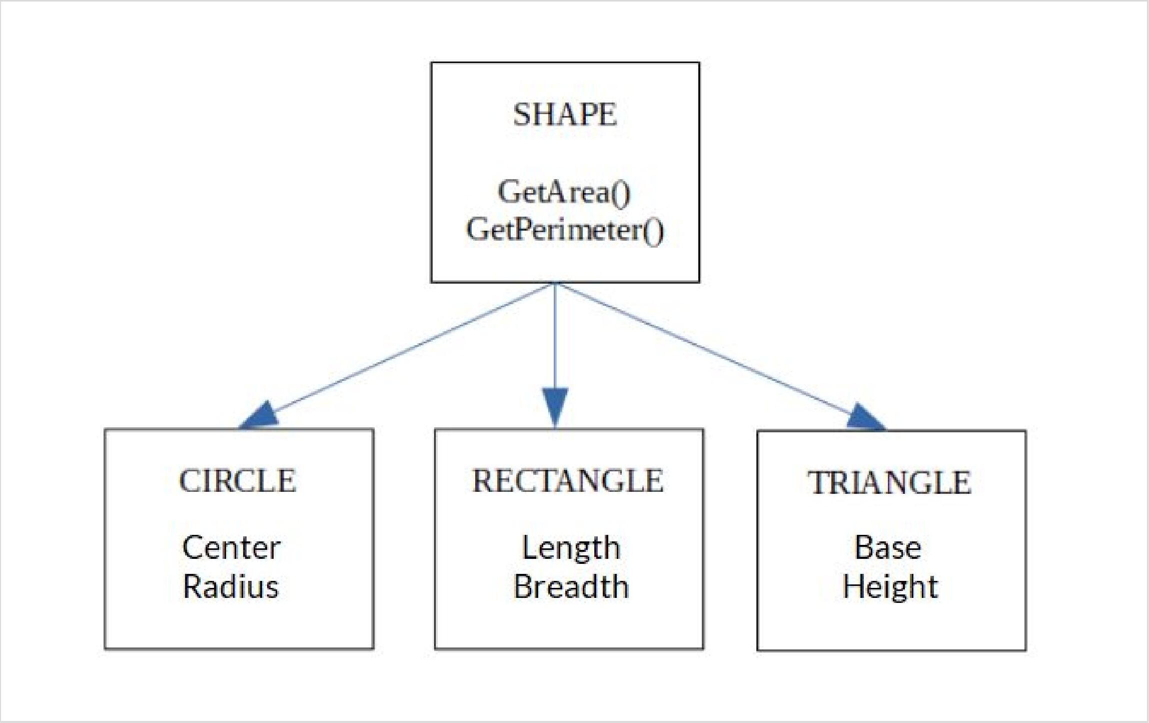
Similar attributes and methods are grouped together using a class. An object can be called as an instance of the class.

Inheritance

A new class can be derived from the original class. The derived class contains attributes and methods of the original class as well as its own.

Example

An Example of the Object Oriented data model is −



Shape, Circle, Rectangle and Triangle are all objects in this model.

Circle has the attributes Center and Radius.

Rectangle has the attributes Length and Breadth.

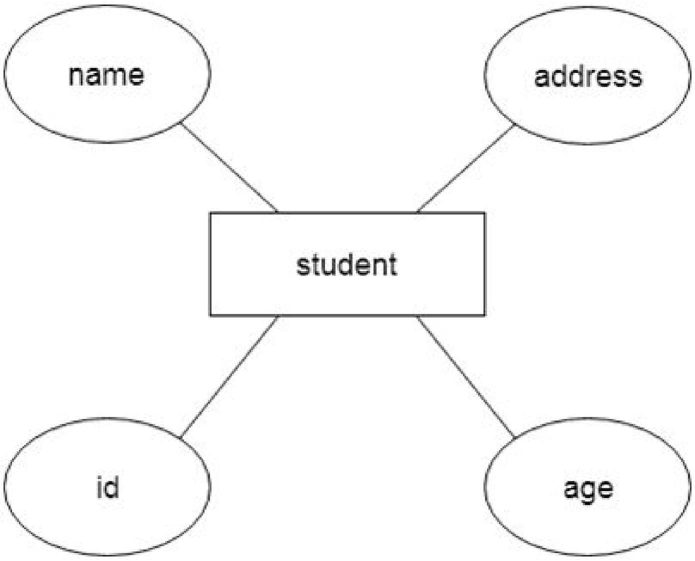
Triangle has the attributes Base and Height.

The objects Circle, Rectangle and Triangle inherit from the object Shape.

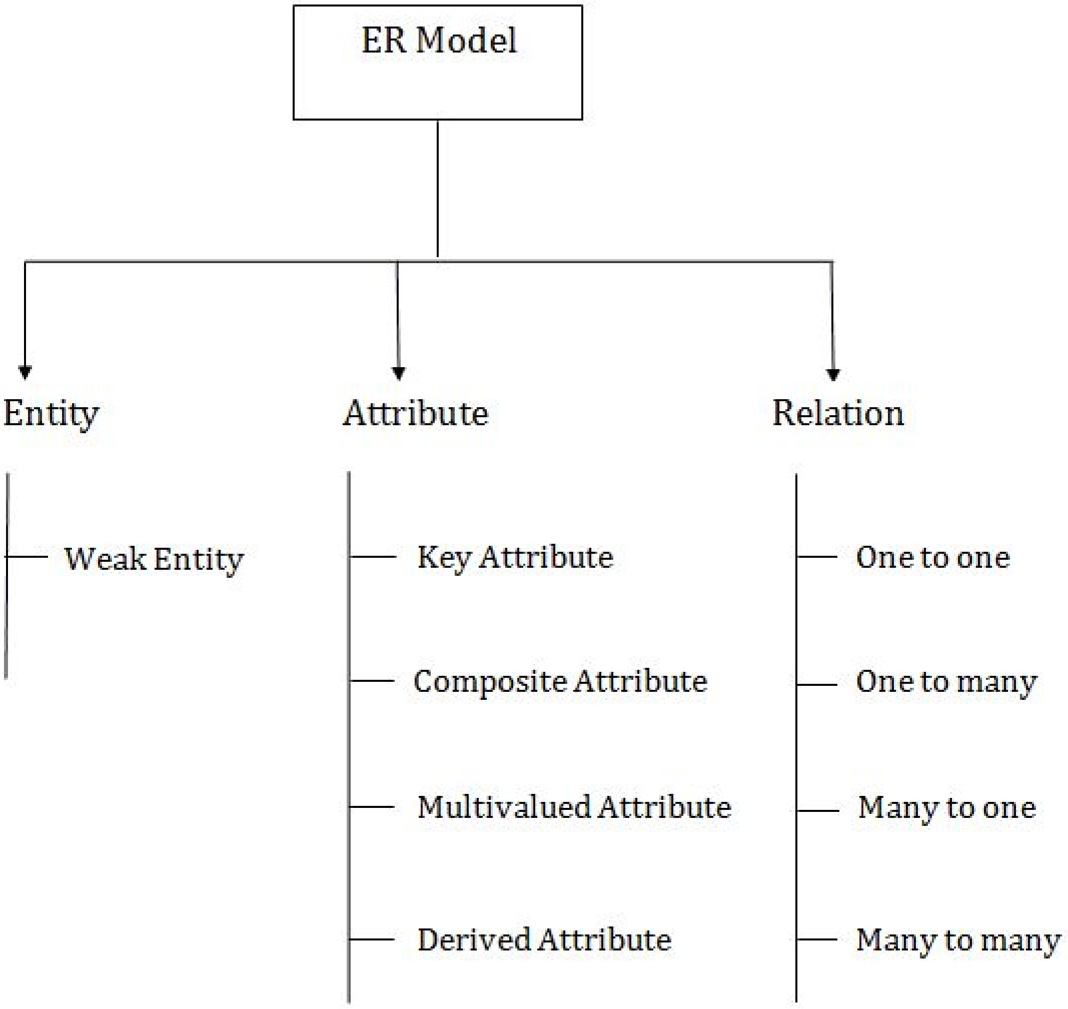
Entity-Relationship Data Model: An ER model is the logical representation of data as objects and relationships among them. These objects are known as entities, and relationship is an association among these entities. This model was designed by Peter Chen and published in 1976 papers. It was widely used in database designing. A set of attributes describe the entities. For example, student_name, student_id describes the 'student' entity. A set of the same type of entities is known as an 'Entity set', and the set of the same type of relationships is known as 'relationship set'.ER model stands for an Entity-Relationship model. It is a high-level data model. This model is used to define the data elements and relationship for a specified system.

- It develops a conceptual design for the database. It also develops a very simple and easy to design view of data.
- In ER modeling, the database structure is portrayed as a diagram called an entity-relationship diagram.

For example, Suppose we design a school database. In this database, the student will be an entity with attributes like address, name, id, age, etc. The address can be another entity with attributes like city, street name, pin code, etc and there will be a relationship between them.



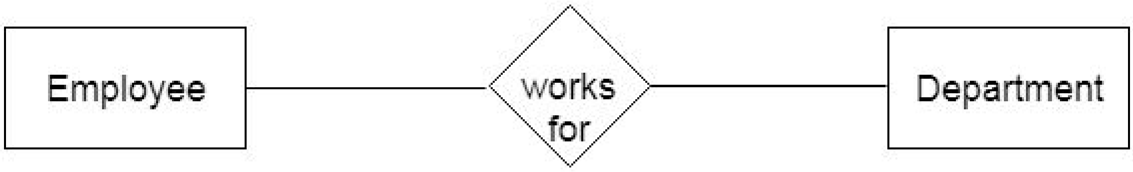
Component of ER Diagram



1. Entity:

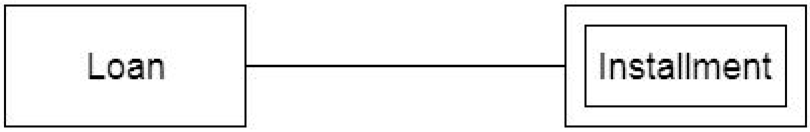
An entity may be any object, class, person or place. In the ER diagram, an entity can be represented as rectangles.

Consider an organization as an example- manager, product, employee, department etc. can be taken as an entity.



a. Weak Entity

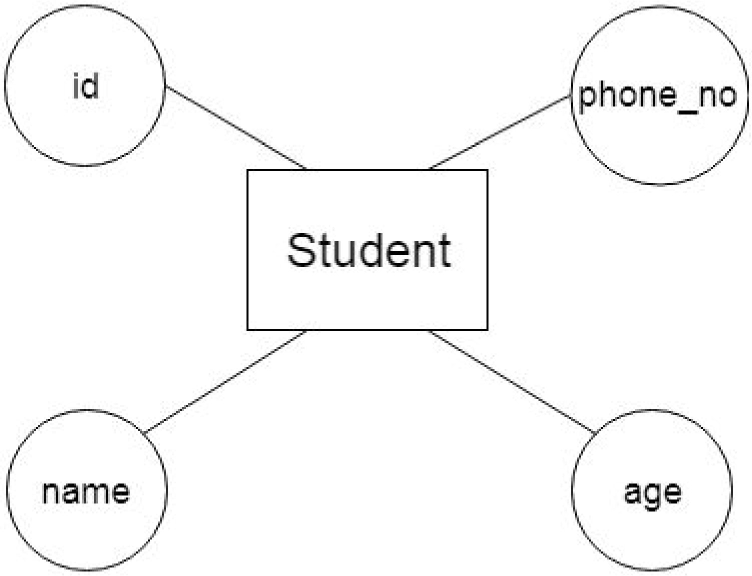
An entity that depends on another entity called a weak entity. The weak entity doesn't contain any key attribute of its own. The weak entity is represented by a double rectangle.



2. Attribute

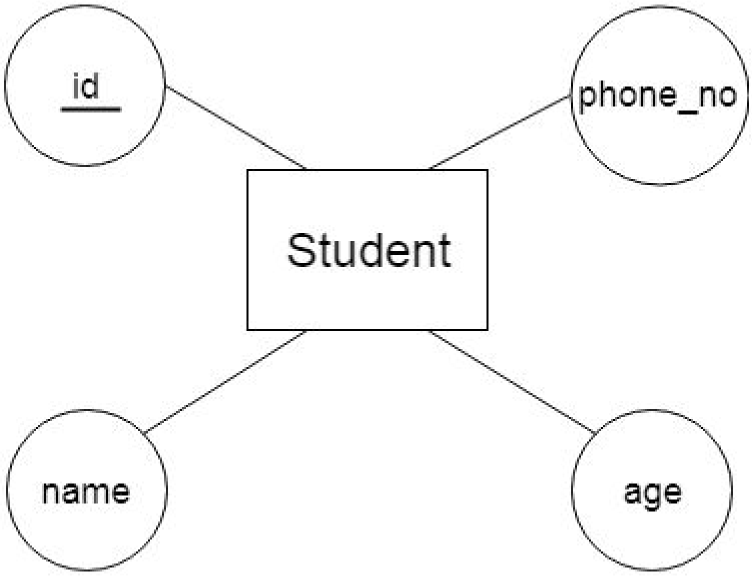
The attribute is used to describe the property of an entity. Eclipse is used to represent an attribute.

For example, id, age, contact number, name, etc. can be attributes of a student.



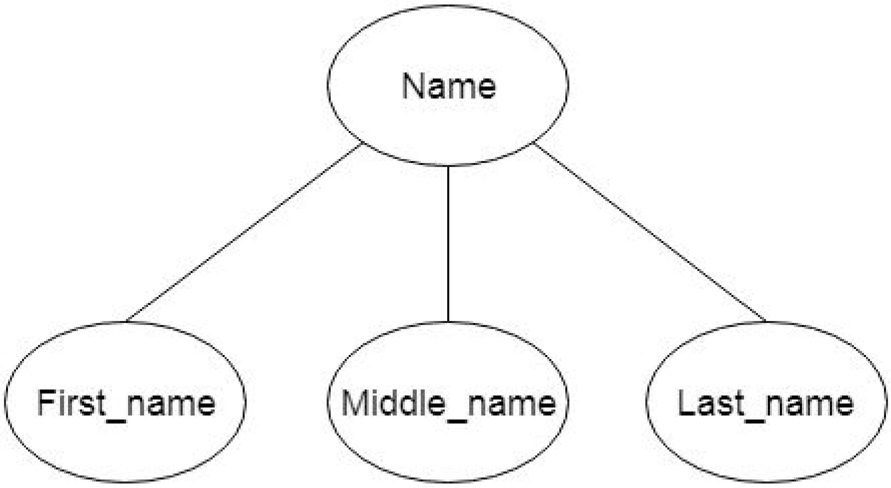
a. Key Attribute

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.



b. Composite Attribute

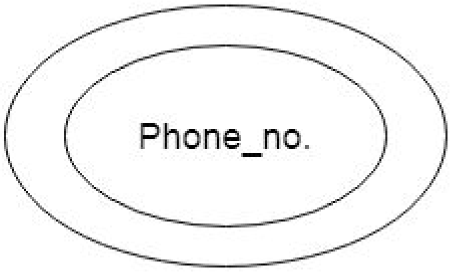
An attribute that composed of many other attributes is known as a composite attribute. The composite attribute is represented by an ellipse, and those ellipses are connected with an ellipse.



c. Multivalued Attribute

An attribute can have more than one value. These attributes are known as a multivalued attribute. The double oval is used to represent multivalued attribute.

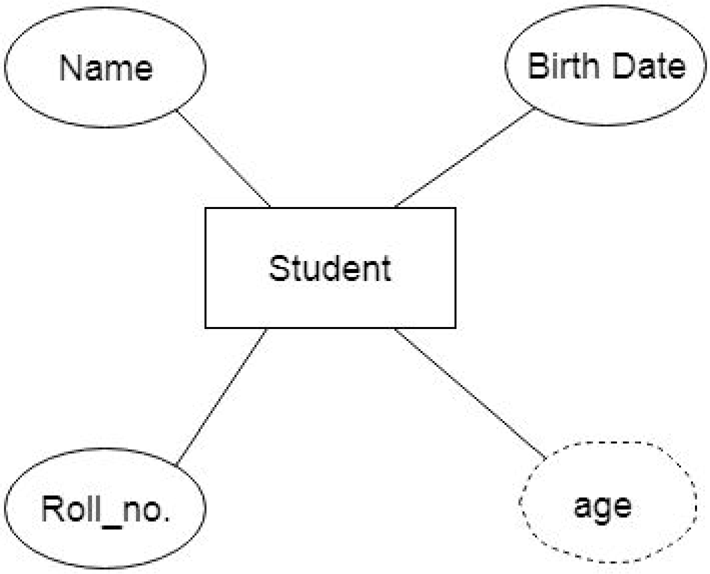
For example, a student can have more than one phone number.



d. Derived Attribute

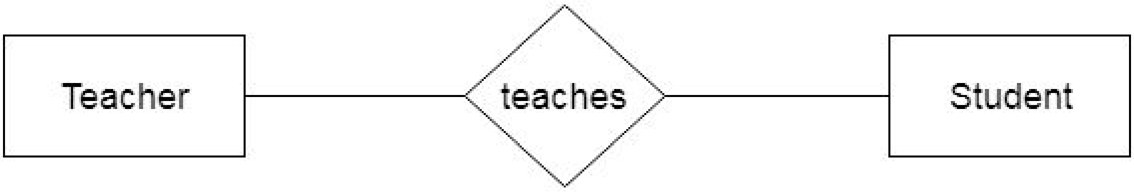
An attribute that can be derived from other attribute is known as a derived attribute. It can be represented by a dashed ellipse.

For example, A person's age changes over time and can be derived from another attribute like Date of birth.



3. Relationship

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

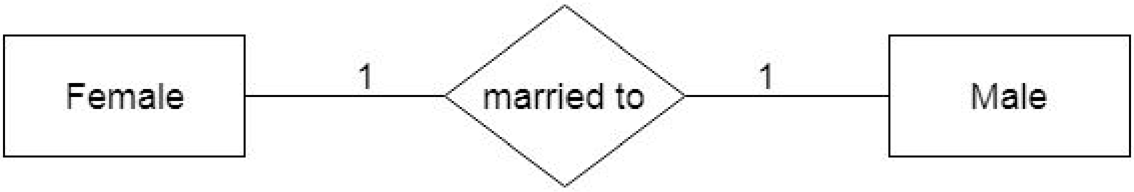


Types of relationship are as follows:

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.

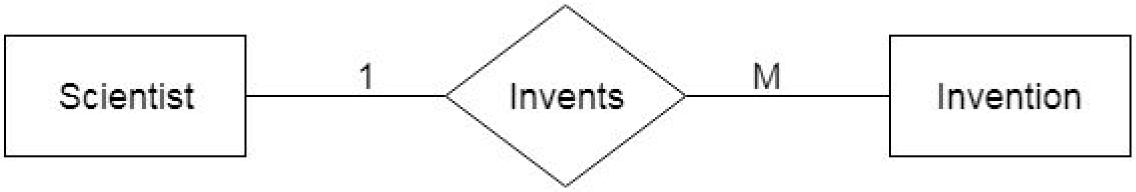
For example, A female can marry to one male, and a male can marry to one female.



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.

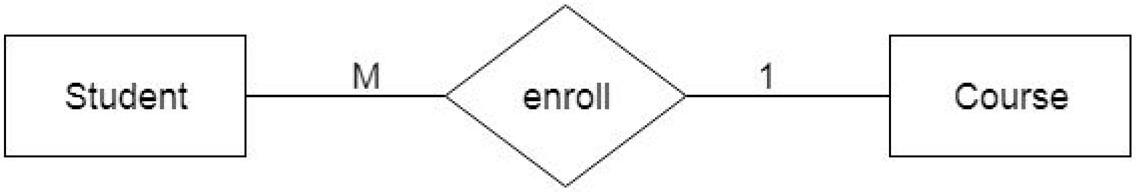
For example, Scientist can invent many inventions, but the invention is done by the only specific scientist.



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.

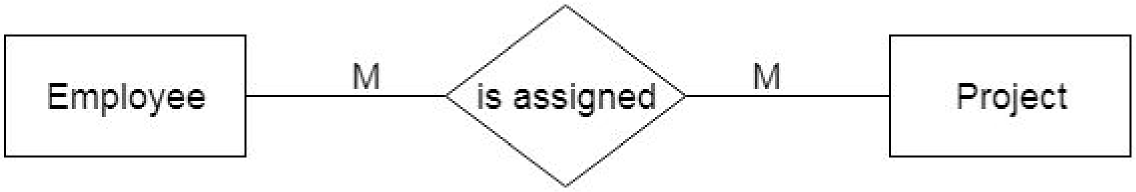
For example, Student enrolls for only one course, but a course can have many students.



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.

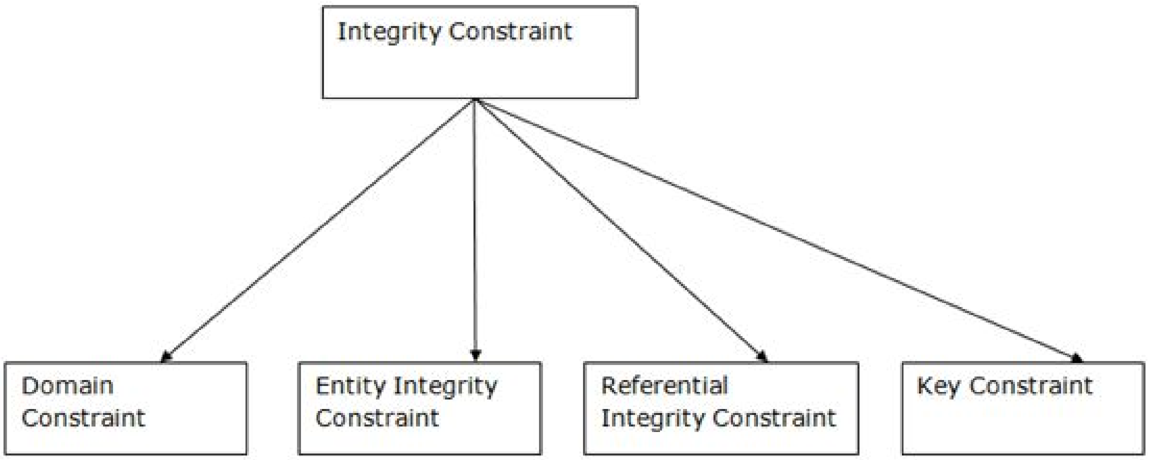
For example, Employee can assign by many projects and project can have many employees.



Integrity Constraints

- Integrity constraints are a set of rules. It is used to maintain the quality of information.
- Integrity constraints ensure that the data insertion, updating, and other processes have to be performed in such a way that data integrity is not affected.
- Thus, integrity constraint is used to guard against accidental damage to the database.

Types of Integrity Constraint

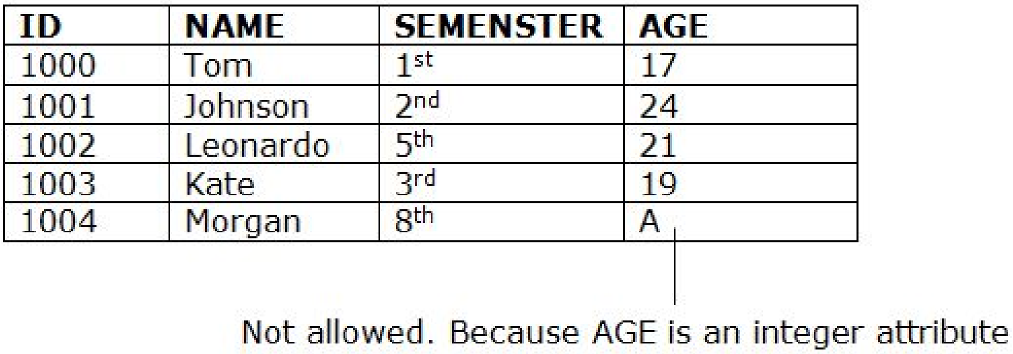


1. Domain constraints

- Domain constraints can be defined as the definition of a valid set of values for an attribute.

- The data type of domain includes string, character, integer, time, date, currency, etc. The value of the attribute must be available in the corresponding domain.

Example:



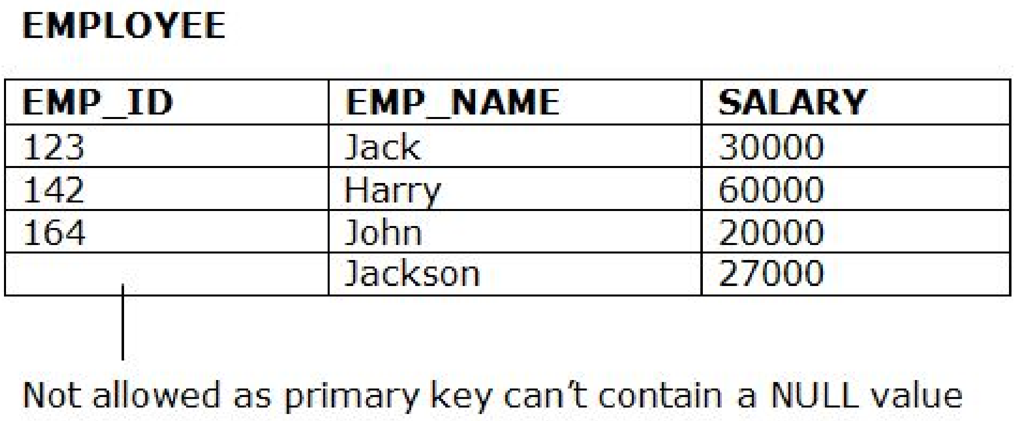
ID	NAME	SEMENSTER	AGE
1000	Tom	1st	17
1001	Johnson	2nd	24
1002	Leonardo	5th	21
1003	Kate	3rd	19
1004	Morgan	8th	A

Not allowed. Because AGE is an integer attribute

2. Entity integrity constraints

- The entity integrity constraint states that primary key value can't be null.
- This is because the primary key value is used to identify individual rows in relation and if the primary key has a null value, then we can't identify those rows.
- A table can contain a null value other than the primary key field.

Example:



EMPLOYEE

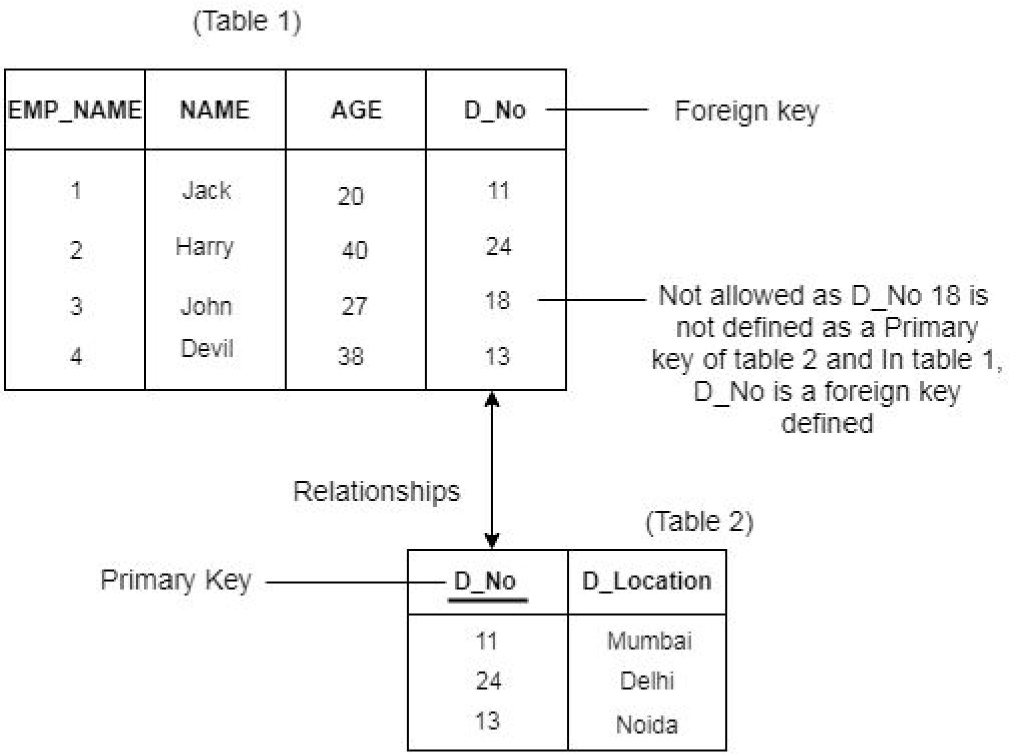
EMP_ID	EMP_NAME	SALARY
123	Jack	30000
142	Harry	60000
164	John	20000
	Jackson	27000

Not allowed as primary key can't contain a NULL value

3. Referential Integrity Constraints

- A referential integrity constraint is specified between two tables.
- In the Referential integrity constraints, if a foreign key in Table 1 refers to the Primary Key of Table 2, then every value of the Foreign Key in Table 1 must be null or be available in Table 2.

Example:



(Table 1)

EMP_NAME	NAME	AGE	D_No
1	Jack	20	11
2	Harry	40	24
3	John	27	18
4	Devil	38	13

(Table 2)

D_No	D_Location
11	Mumbai
24	Delhi
13	Noida

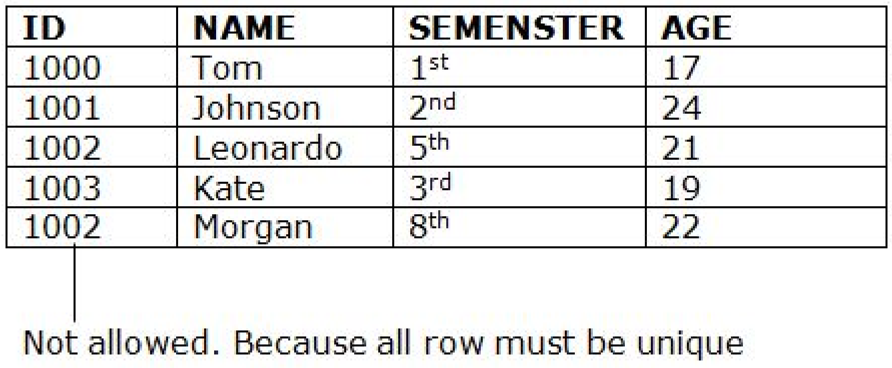
Foreign key
Not allowed as D_No 18 is not defined as a Primary key of table 2 and In table 1, D_No is a foreign key defined
Relationships
Primary Key

4. Key constraints

- Keys are the entity set that is used to identify an entity within its entity set uniquely.

- An entity set can have multiple keys, but out of which one key will be the primary key. A primary key can contain a unique and null value in the relational table.

Example:



ID	NAME	SEMENSTER	AGE
1000	Tom	1st	17
1001	Johnson	2nd	24
1002	Leonardo	5th	21
1003	Kate	3rd	19
1002	Morgan	8th	22

Not allowed. Because all row must be unique

UNIT - 1 COMPLETED

For more content visit our website : https://cgccollegespace.live

For updates visit our Instagram profile -- https://www.instagram.com/cgccollegespace/