Unit-1
Database Management System

Data:
The collection of raw facts or unprocessed means that can be found after some experiments or observations and need to be processed is called data. They act as input and can be processed to produce information.
E.g.: Ram, a, student.

Information:
The meaningful result which is obtained after processing the data is called information. It is the output of data processing and depends on data.
E.g.: Ram is a student.

Data Processing:
The process of converting unprocessed means(i.e. data) into meaningful information is known as data processing.

Database:
The organized collection of logically related data relevant to an individual or an enterprise is called a database. It is a repository or a container where data is stored in an organized manner.
E.g.: Telephone Directory, Result sheet, Customer Records, etc.

Database System:
Database system is a collection of interrelated data and a set of application programs required to access, update and manage the data.
The major uses of database application software are:

- ➔ Storing and Accessing data
- ➔ Updating or modifying data
- ➔ Deleting data
- ➔ Sharing of data for concurrent access
- ➔ Enforcing standards or rules for accessing the data
- ➔ Reducing data redundancy or duplication
- ➔ Maintaining accuracy of data

Database Management System (DBMS):
A set of programs that enables users to create, manage and manipulate databases is called database management system. DBMS acts as an interface between database and

user for the manipulation (i.e., adding, retrieving, modifying and deleting the records) of the database.
E.g.: MS-Access, Oracle, MySQL, dBase, FoxPro, PostgreSQL, SQL Server, etc.

Objectives of DBMS:

- ➔ Making easy access to data for users
- ➔ Eliminate redundant data from database
- ➔ Allow multiple users to access the database at a time
- ➔ Provide data security from unauthorized access
- ➔ Provide a quick response to the user's request to access data

Advantages of DBMS:

DBMS provides lots of advantages over traditional flat file system which are as follows:

- ➔ **Reduce data redundancies:**
  Redundancy is the concept of repetition of data (i.e., each data may have more than a single copy). DBMS overcomes this issue and reduces the data redundancy.
- ➔ **Data sharing:**
  DBMS allows the data stored in the database to be shared among the end-users and provides concurrent access to data.
- ➔ **Data Independence:**
  DBMS keeps data separate from application programs. Thus, any change can be made in data without changing application programs. It means data independence exists.
- ➔ **Maintains integrity:**
  DBMS ensures that the data stored in a database is valid and accurate. It avoids data inconsistency and hence the data integrity is maintained.
- ➔ **Efficient data access:**
  Data is stored in an organized manner in the database and hence it is easy to access the data for the users in an efficient way.
- ➔ **Removes data inconsistency:**
  The changes made on duplicated data causes data inconsistency. DBMS reduces the data inconsistency by eliminating two entries regarding the same data.
- ➔ **Improved security:**
  DBMS sets some restrictions or privileges to the users on accessing the data providing different levels of data abstraction. This improves data security.

Disadvantages of DBMS:

Although the database management system has considerable advantages, it has some disadvantages too which are as follows:

- ➔ **Increased cost:**
  It requires high initial investment to implement and requires skilled manpower to operate the database system which increases the cost.
- ➔ **Managing complexities:**
  It is not easy to manage and maintain a database system. Designing a fully functional database is itself a challenging task. So, it requires special manpower having the knowledge of databases to manage it.
- ➔ **Frequent Updates:**
  The frequent changing technologies lead to the updates on existing systems adding new functionality on database systems. This requires a system update which adds cost.
- ➔ **Performance:**
  The size of a database increases when a huge amount of data is stored on it and the performance of the database system decreases because of increased data size stored on the database.
- ➔ **Database Failure:**
  In a database system, all the data is stored in the database. The failure of the database leads to data loss which is a big problem with DBMS.

Database Model:

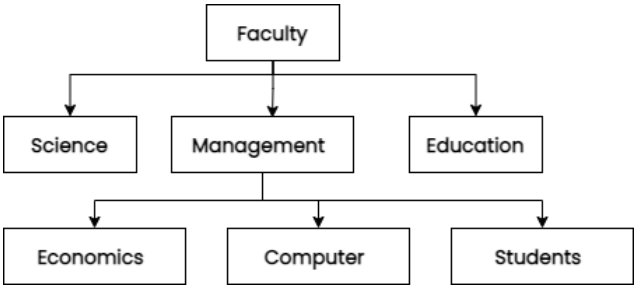
A conceptual tool for describing data, relationships among data, data semantics, etc. is called database model. It describes the underlying structure of the database.

The different types of database models are as follows:

1. Hierarchical Database Model
2. Network Database Model
3. Relational Database Model
4. Entity Relationship Model
5. Object Oriented Model

Hierarchical Database Model:

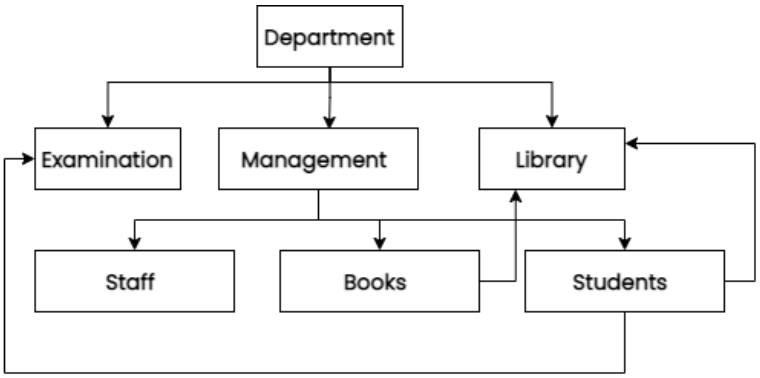
The database model in which the records are organized in hierarchical manner or top-down structure like a branch of a tree is known as hierarchical database model.



The top-level record is called root whereas the bottom level records are called leaves. The records which are in between root and leaves are called intermediate records.

Network Database Model:

The database model in which multidimensional connections exist among the data (i.e., each child can be linked with more than one parent) such that it can be accessed from more than one parent is known as network database model. It provides more flexibility than hierarchical database model.



Relational Database Model:

The database model in which a database is represented as the collection of relation, where relation is represented by a two-dimensional table is known as relational database model. In this model, data are arranged in tabular form. It is the most common database model.

StudentID	StudentName	Address	Phone
101	Samriddhi Sakya	Chitwan	9841223081
102	Chetan Rawat	Syanga	9812347890
103	Nikita Sharma	Ghorahi	9854123745

Terminologies related to Relational Database Model:

- ➔ **Table:**
  A table is a collection of rows and columns where data in a database is stored. It is also called relation.
- ➔ **Domain:**
  A domain is a set of permissible values that can be given to an attribute or field of an entity. Every field or attribute in a table has a specific domain so that values to these fields or attributes cannot be assigned outside their domain.

➔ **Attributes:**

Attributes are the properties which describe the entity. They are also called elements or fields of a table. The attributes can be of various types as follows:

- ★ **Simple Attribute** - Attribute that cannot be further divided. E.g.: First name, age, etc.
- ★ **Composite Attribute** - Attribute that can be further divided. E.g.: Name, address, etc.
- ★ **Single Valued Attribute** - Attribute having a single value for an entity. E.g.: Last name, age, etc.
- ★ **Multi-valued Attribute** - Attribute having multiple values for an entity. E.g.: Mobile number
- ★ **Stored Attribute** - Attribute whose values are physically stored in a database. E.g.: StudentID, studentName, etc.
- ★ **Derived Attribute** - Attribute whose values can be derived from the values of other related attributes. E.g.: Age

➔ **Tuple:**

Each row of a relation(i.e., table) called tuple. It is also called record.

Entity Relationship(ER) Model:

The database model that describes the design of the database in terms of entities and relationships among them. It is a high level conceptual data model.

Terminologies related to Entity Relationship Database Model:

➔ **Entity:**

An entity is a thing or an object in the real world that is distinguishable from another object.

➔ **Relationship:**

An association among several entities is called a relationship. For example, a depositor relationship associates a customer with each account he/she has.

The relationship that can exist between any two entity sets A and B may be:

- ★ **One to one** - A relationship is said to be one to one if an entity in one set associates with at most one entity in another set.

  For example: Relationship between husband & wife, relationship between college and principal, etc.
- ★ **One to many** - A relationship is said to be one to many if an entity in one set associates with multiple entities in another set.

  For example: Relationship between college & students, relationship between college & faculties, etc.
- ★ **Many to one** - A relationship is said to be many to one if multiple entities of one set associate with at most one entity in another set.

For example: Relationship between contributors & project relationship between authors & book, etc.

- ★ **Many to many** - A relationship is said to be many to many if multiple entities of one set associate with multiple entities in another set.
  For example: Relationship between student & course, relationship between teacher and students, etc.

➔ **Entity set and Relationship set:**

A set of entities of the same type that share the same properties is called entity set. For example: set of all customers, set of all courses in a university.

A set of all relationships of the same type is called a relationship set.

➔ **Attribute:**

Attribute is a property that describes an entity set.

For example: customerID, customerName, customerAddress, etc are the attributes of entity Customer.

➔ **Domain:**

The set of permitted values for an attribute is called domain of that attribute. For example: set of all strings of certain length for CustomerName is the domain of that attribute.

➔ **ER Diagram:**

A graphical representation of entities and their relationships is called entity relationship (ER) diagram. It has the capability to express the overall logical structure of a database graphically.

The different shapes are used to represent the database graphically as follows:

- ★ Rectangle - represents entity set
- ★ Ellipse - represents attributes
- ★ Diamond - represents relationship sets
- ★ Line - links attributes to entity sets & entity sets to relationship sets
- ★ Double Ellipse - represents multivalued attributes
- ★ Dashed Ellipse - represents derived attributes
- ★ Underline - indicates primary key attributes

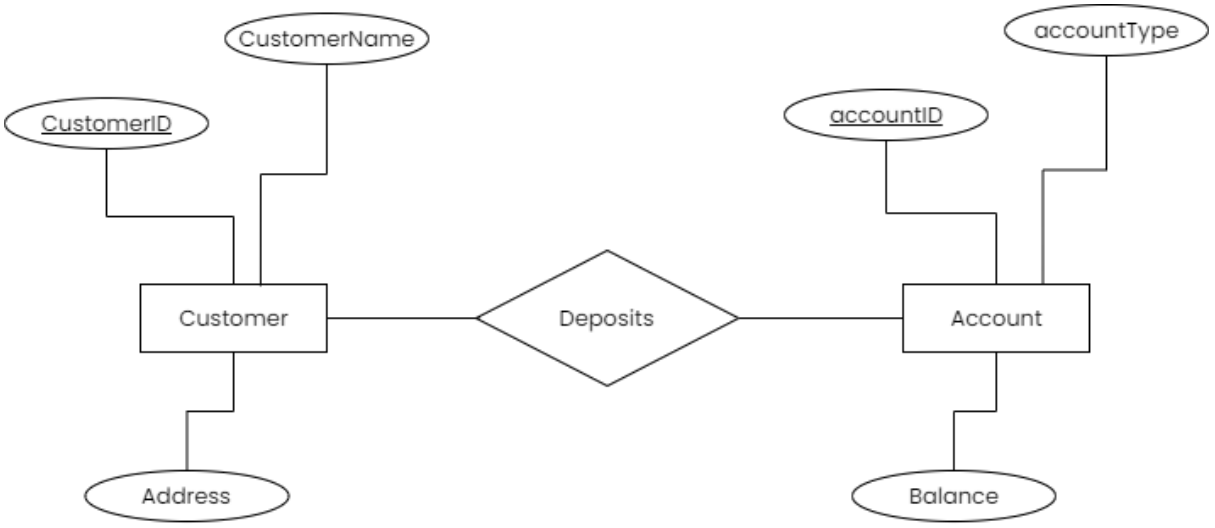


Fig.: ER Diagram representing the relationship between customer and account

Object-Oriented Model:

The database model based on the collection of objects with the notion of encapsulation, methods and object identity is known as object-oriented model. Objects that contain the same type of values and the same methods are grouped into classes.

For example: Consider an object representing a bank account. The object may contain instance variable accountNumber and balance.

Relational Database Management System (RDMS):

The database system which is used to manage and manipulate the data in the relational database model is called relational database management system or RDMS. It stores data in the different tables and the relationship between the tables can be formed by using a common field.

The examples of relational database management system software are SQL server, Oracle, MySQL, etc.

Keys:

An attribute which is used to identify a particular record in a database is called a key. It may be a single attribute or composed of more than one attribute. Any attribute that is a part of a key is known as key attribute.

The different types of keys used in database are:

- **Candidate Key:**

  Any attribute or the combination of attributes that can uniquely identify the records of a database is called candidate key.

- **Primary Key:**
  A candidate key that can uniquely identify every row in a database table is called primary key. Only one candidate key is selected as a primary key.
- **Foreign Key:**
  A key attribute that is used to create the relationship between two tables is called foreign key.
- **Super Key:**
  A set of all the keys that helps to identify rows in a table uniquely is called super key. It is the superset of a candidate key.
- **Alternate Key:**
  A candidate key which is not used as a primary key is called an alternate key.

Normalization:

A systematic approach of decomposing tables organizing the data into multiple related tables to eliminate data redundancy is known as normalization. It is an approach for designing reliable database systems.

Repetition of the same data in multiple places not only increases the size of the database but also leads to data anomalies.

Data anomalies:

The inconsistencies in the data stored in a database due to duplication of same data in multiple locations resulting problems on inserting, deleting and updating records are called data anomalies.

The different types of data anomalies are:

- Insertion Anomaly
- Updation Anomaly
- Deletion Anomaly

Need of Normalization:

There is considerable amount of data redundancy and some serious problems on inserting, updating and deletion of records in a database. Thus, to reduce the data redundancy as far as possible and for easy insert, delete and update operation in relational database systems removing data inconsistencies we need normalization.

Advantages of Normalization:

Normalization offers lots of advantages among which some of them are listed below:

- Reduces data redundancy.
- Simplifies the structure of the database table.
- Avoids loss of information.
- Makes it easy to insert, update and delete records.
- Removes data inconsistencies.

Types of Normal Forms:

Normalization theory is built under the concept of normal forms. The different normal forms are:

a) First Normal Form (1NF)
b) Second Normal Form (2NF)
c) Third Normal Form (3NF)
d) Fourth Normal Form (4NF)
e) Fifth Normal Form (5NF)

Non-Normalized Relation:

A relation is said to be a non-normalized relation if any domain of attributes of that relation are non-atomic.

RollNo	StudentName	Class	SubjectName	Marks
1	Naresh	11	English	78
			Physics	89
			Chemistry	70
5	Alish	12	Computer	90
			Nepali	85

Fig-1.: Non-Normalized Relation

First Normal Form (1NF):

A relation is said to be in first normal form if the domain of all attributes of that relation are atomic (i.e., there should not be any repeating groups of an attribute). The purpose of 1NF is to eliminate repeating groups of attributes in an entity.

Example: The non-normalized relation shown in Fig-1 can be represented in 1NF as:

RollNo	StudentName	Class	SubjectName	Marks
1	Naresh	11	English	78
1	Naresh	11	Physics	89
1	Naresh	11	Chemistry	70
5	Alish	12	Computer	90
5	Alish	12	Nepali	85

Table-1: Relation in 1NF

Second Normal Form (2NF):

A relation is said to be in second normal form if it is in first normal form and should not contain partial dependency. The purpose of the second normal form is to remove the partial key dependencies.

In Table-1, the primary key = combination of RollNo & Class.

Here, StudentName depends on RollNo & Class, SubjectName depends only on Class whereas Marks depends on SubjectName & StudentName. So, Table-1 in 1NF can be decomposed on following tables:

StudentName	RollNo	Class
Naresh	1	11
Alish	5	12

Table-1: Student Table

SubjectName	Class
English	11
Physics	11
Chemistry	11
Computer	12
Nepali	12

Table-2: Subject Table

StudentName	SubjectName	Marks
Naresh	English	78
Naresh	Physics	89
Naresh	Chemistry	70
Alish	Computer	90
Alish	Nepali	85

Table-3: Marks Table

The tables are in 2NF.

Third Normal Form (3NF):

A relation is said to be in third normal form if it is second normal form and should not contain transitive dependency on primary key.

Example: The above table in 2NF can be represented in 3NF by decomposing into following tables:

StudentId	StudentName	RollNo	ClassId
S01	Naresh	1	C01
S02	Alish	5	C02

Table-1: Student Table

SubjectId	SubjectName
100	English
101	Physics
102	Chemistry
103	Computer
104	Nepali

Table-2: Subject Table

StudentId	SubjectId	Marks
S01	100	78
S01	101	89
S01	102	70
S02	103	90
S02	104	85

Table-3: Marks Table

ClassId	Class
C01	11
C02	12

Table-4: Class Table

Structured Query Language (SQL):
SQL is a query language which has a number of statements through which the user requests information from the database. It is the universal standard database language for accessing and managing data in the database.

Types of Database Language:

- ➔ **Data Definition Language (DDL):**
  A database language used by database designers and programmers to specify the database schema(i.e., logical structure of a database) is called data definition language.
- ➔ **Data Manipulation Language (DML):**
  A database language that enables the users to access or manipulate the data in a database is called data manipulation language.

Centralized and Distributed Database System:

1. **Centralized Database System:**
   The database system in which data is stored in a single database is called a centralized database system. There is a single database server to store all the data and files, making it accessible to other clients in the network.
   Advantages:
   - ➔ Simple to implement & manage.
   - ➔ Low cost to set up.
   - ➔ Provides high security.
   - ➔ Suitable for small organizations with limited data and users.

   Disadvantages:
   - ➔ Server failure leads to the whole system down.
   - ➔ Less reliable.
   - ➔ Performance decreases on increasing number of users.
   - ➔ Not suitable for large organizations.
2. **Distributed Database System:**
   The database system system in which data is stored in a distributed database that facilitates easy access to data from any location is called a distributed database system.
   Advantages:
   - ➔ Failure of one server doesn't affect the whole system.
   - ➔ More reliable than a centralized database system.
   - ➔ Improved availability of data
   - ➔ Suitable for large organizations with a large number of users.

   Disadvantages:
   - ➔ Complex to implement & manage.
   - ➔ Less data security.
   - ➔ High cost of implementation and operation.
   - ➔ More complexity on database design.

Data Abstraction:
A mechanism to hide the complexity of a database from the users is known as data abstraction. It hides how data are actually stored and maintained in a database.

There are three level of abstraction as mentioned below:

a. **Physical Level** - It is the lowest level of abstraction which describes how actually the data are stored in a database.
b. **Logical Level** - It is a higher level of abstraction than physical level which describes what data are stored in a database and what relationships exist among them.
c. **View Level** - It is the highest level of abstraction which describes only a part of the entire database. It allows each user/application to get a different perspective of the database.

Data Security:

The protection of data from unauthorized access and accidental or intentional loss of data is known as data security. Not every user of the database system should be able to access all data.

The data security can be achieved in the following ways:

- ➔ By doing regular backup of data stored in the database.
- ➔ By using passwords to restrict unauthorized users.
- ➔ By using an Intrusion-detection system.
- ➔ By applying cryptographic tools, firewalls on the system.

Data Integrity:

Data integrity or integrity constraint is a rule that restricts the values that may be present in the database. It refers to the overall completeness, accuracy and consistency of data.

There are three types of data integrity:

- ➔ **Domain Integrity:**
  It is an integrity constraint that specifies a valid set of values for an attribute. It may also allow the use of NULL values for particular attributes.
- ➔ **Entity Integrity:**
  It is an integrity constraint which ensures that the primary key value can't be NULL. It is maintained in the database through the primary key.
- ➔ **Referential Integrity:**
  It is an integrity constraint which tells that 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.

Database Administrator (DBA):

The database administrator is a person having central control over data and programs accessing that data & is responsible for maintaining the DBMS in an organization.

Responsibilities of DBA:

Database Administrator (DBA) has the following responsibilities:

- ➔ **Defining Schema:**
  DBA is responsible for defining both internal schema (i.e., physical database design) and external schema (i.e., logical database design).

- **Storage structure and access method definition:**
  DBA is responsible for writing a set of definitions to define storage and access methods.
- **Granting authorization for data access:**
  DBA is responsible for granting different types of authorization for data access to various users.
- **Defining security and integrity rules:**
  DBA is responsible for defining security rules to protect data from unauthorized users and integrity rules for checking accuracy or validity of data.
- **Routine Maintenance:**
  DBA is responsible for periodically backing up the database ensuring free disk space & monitoring performance and tuning accordingly.