Database - Introduction

The word data refers to information or facts usually collected as the result of experience , observation or experiment or process within a computer system.

Database pays an important role in all the areas where computers are included such as library , education, medicine , science etc.

DATA :- it is a term which is very simple to grab. Data is defined as collection of meaningful facts which can be stored and processed by the human or computer. In other words data is a collection of unorganized facts , concepts or instructions in a formalized manner suitable for interpretation, processing by human or computer system. Data can be number , alphabet, words , special symbols and etc.

INFORMATION :- when the data is processed and converted into a meaningful and useful form, it is known as information. It will be generated after the arranging the data into a suitable and meaningful form. Traditionally, the data was stored in voluminous repositories such as files, books and ledgers.

Difference between Data and Information

DATA	INFORMATION
1) It is collection of unorganized facts.	2)It is the intelligent form of data
2) It is Facts, statistics used for reference or analysis	2) It is Knowledge derived from study, experience or instructions.
3) Numbers, characters, symbols, images etc which can be processed by a computer.	3) It is a tool i.e. Communication of intelligence.
4) Data must be interpreted, by a human or machine, to derive meaning.	4) Information i any kind of knowledge that is exchangeable amongst people , about things, facts, concepts, etc , in some context.
5) Data is representation of information	5) Information is interpreted data.

DATABASE

A **database** is an organized collection of data. It is the collection of schemas, tables, queries, reports, views and other objects. The data are typically organized to model aspects of reality in a way that supports processes requiring information, such as modelling the availability of rooms in hotels in a way that supports finding a hotel with vacancies.

A **database management system** (**DBMS**) is a computer software application that interacts with the user, other applications, and the database itself to capture and analyze data. A general-purpose DBMS is designed to allow the definition, creation, querying, update, and administration of databases. Well-known DBMS include MySQL, Microsoft SQL Server, Oracle. Formally, a "database" refers to a set of related data and the way it is organized. Access to these data is usually provided by a "database management system" (DBMS) consisting of an integrated set of computer software that allows users to interact with one or more databases and provides access to all of the data contained in the database (although restrictions may exist that limit access to particular data.

Characteristics of database

1) Data present is correct.
2) Data stored is flexible to change
3) There is no duplicate data present
4) Data is permanent in database
5) Data is secured and protected
6) Data should be independent in the database
7) Data is well organized
8) Data is be available when needed
9) Data in the database is shared by different users and applications.

Advantages of a DBMS

Using a DBMS to manage data has many advantages:

1) Data independence: Application programs should be as independent as possible from details of data representation and storage. The DBMS can provide an abstract view of the data to insulate application code from such details.

2) Efficient data access: A DBMS utilizes a variety of sophisticated techniques to store and retrieve data efficiently. This feature is especially important if the data is stored on external storage devices.

3) Data integrity and security: If data is always accessed through the DBMS, the DBMS can enforce integrity constraints on the data. For example, before inserting salary information for an employee, the DBMS can check that the department budget is not exceeded. Also, the DBMS can enforce access controls that govern what data is visible to different classes of users.

4) Data administration: When several users share the data, centralizing the administration of data can offer significant improvements. Experienced professionals who understand the nature of the data being managed, and how different groups of users use it, can be responsible for organizing the data representation to minimize redundancy and finetuning the storage of the data to make retrieval efficient.

5) Concurrent access and crash recovery: A DBMS schedules concurrent accesses to the data in such a manner that users can think of the data as being accessed by only one user at a time. Further, the DBMS protects users from the effects of system failures.

Disadvantages of a DBMS

1) **Danger of a Overkill:** For small and simple applications for single users a database system is often not advisable.
2) **Complexity**: A database system creates additional complexity and requirements. The supply and operation of a database management system with several users and databases is quite costly and demanding.
3) **Qualified Personnel**: The professional operation of a database system requires appropriately trained staff. Without a qualified database administrator nothing will work for long.
4) **Costs:** Through the use of a database system new costs are generated for the system itselfs but also for additional hardware and the more complex handling of the system
5) **Lower Efficiency**: A database system is a multi-use software which is often less efficient than specialised software which is produced and optimised exactly for one problem

The Three-Schema Architecture

In this architecture, schemas can be defined at the following three levels:

1. **The internal level** has an internal schema, which describes the physical storage structure of the database. The internal schema uses a physical data model and describes the complete details of data storage and access paths for the database.
2. **The conceptual level** has a conceptual schema, which describes the structure of the whole database for a community of users. The conceptual schema hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user operations, and constraints. A high-level data model or an implementation data model can be used at this level
3. **The external or view level** includes a number of external schemas or user views. Each external schema describes the part of the database that a particular user group is interested in and hides the rest of the database from that user group. A high-level data model or an implementation data model can be used at this level

Data Independence

The three-schema architecture can be used to explain the concept of data independence, which can be defined as the capacity to change the schema at one level of a database system without having to change the schema at the next higher level. We can define two types of data independence:

1) **Logical data independence** : It is the capacity to change the conceptual schema without having to change external schemas or application programs. We may change the conceptual schema to expand the database (by adding a record type or data item), or to reduce the database (by removing a record type or data item). In the latter case, external schemas that refer only to the remaining data should not be affected. Only the view definition and the mappings need be changed in a DBMS that supports logical data independence. Application programs that reference the external schema constructs must work as before, after the conceptual schema undergoes a logical reorganization. Changes to constraints can be applied also to the conceptual schema without affecting the external schemas or application programs

2) Physical data independence is the capacity to change the internal schema without having to change the conceptual (or external) schemas. Changes to the internal schema may be needed because some physical files had to be reorganized—for example, by creating additional access structures—to improve the performance of retrieval or update. If the same data as before remains in the database, we should not have to change the conceptual schema.

Mapping

The process to convert a request (from external level) and the result between view levels is called mapping. The mapping defines the correspondence between three view levels. The mapping description is also stored in data dictionary. The DBMS is responsible for mapping between these three types of schemas. There are two types of mapping.

(i) External–Conceptual mapping (ii) Conceptual–Internal mapping

1) **External-Conceptual Mapping :**

An external-conceptual mapping defines the correspondence between a particular external view and the conceptual view. The external-conceptual mapping tells the DBMS which objects on the conceptual level correspond to the objects requested on a particular user's external view. If changes are made to either an external view or conceptual view, then mapping must be changed accordingly.

2) **Conceptual-Internal Mapping :**

The conceptual-internal mapping defines the correspondence between the conceptual view and the internal view, i.e. database stored on the physical storage device. It describes how conceptual records are stored and retrieved to and from the storage device. This means that conceptual-internal mapping tells the DBMS that how the conceptual! records are physically represented. If the structure of the stored database is changed, then the mapping must be changed accordingly. It is the responsibility of DBA to manage such changes.

RDBMS

RDBMS stands for Relational Database Management System. RDBMS data is structured in database tables, fields and records. Each RDBMS table consists of database table rows. Each database table row consists of one or more database table fields. **RDBMS** is based on the relational model as invented by E. F. Codd, of IBM's San Jose Research Laboratory.The most popular RDBMS are MS SQL Server, DB2, Oracle and MySQL

Codd’s 12 Rules

These rules can be applied on any database system that manages stored data using only its relational capabilities. This is a foundation rule, which acts as a base for all the other rules.

Rule 1: Information Rule

The data stored in a database, may it be user data or metadata, must be a value of some table cell. Everything in a database must be stored in a **table format.**

Rule 2: Guaranteed Access Rule

Every single data element (value) is guaranteed to be accessible logically with a combination of table-name, primary-key (row value), and attribute-name (column value). No other means, such as pointers, can be used to access data

Rule 3: Systematic Treatment of NULL Values

The NULL values in a database must be given a systematic and uniform treatment. This is a very important rule because a NULL can be interpreted as one the following − data is missing, data is not known, or data is not applicable

Rule 4: Active Online Catalog

The structure description of the entire database must be stored in an online catalog, known as **data dictionary**, which can be accessed by authorized users. Users can use the same query language to access the catalog which they use to access the database itself.

Rule 5: Comprehensive Data Sub-Language Rule

A database can only be accessed using a language having linear syntax that supports data definition, data manipulation, and transaction management operations. This language can be used directly or by means of some application. If the database allows access to data without any help of this language, then it is considered as a violation.

Rule 6: View Updating Rule

All the views of a database, which can theoretically be updated, must also be updatable by the system.

Rule 7: High-Level Insert, Update, and Delete Rule

A database must support high-level insertion, updation, and deletion. This must not be limited to a single row, that is, it must also support union, intersection and minus operations to yield sets of data records.

Rule 8: Physical Data Independence

The data stored in a database must be independent of the applications that access the database. Any change in the physical structure of a database must not have any impact on how the data is being accessed by external applications.

Rule 9: Logical Data Independence

The logical data in a database must be independent of its user's view (application). Any change in logical data must not affect the applications using it. For example, if two tables are merged or one is split into two different tables, there should be no impact or change on the user application. This is one of the most difficult rule to apply.

Rule 10: Integrity Independence

Integrity constraints must be specified separately from application programs and stored in the catalog. it must be possible to change such constraints as and when appropriate without unnecessarily affecting existing applications.

Rule 11: Distribution Independence

The end-user must not be able to see that the data is distributed over various locations. Users should always get the impression that the data is located at one site only. This rule has been regarded as the foundation of distributed database systems.

Rule 12: Non-Subversion Rule

If a system has an interface that provides access to low-level records, then the interface must not be able to subvert the system and bypass security and integrity constraints

e.g. :- Let's say you define a non-null constraint on a column. Can you bypass the RDBMS to insert a null in that column? If so, you've violated that rule

Difference between DBMS and RDBMS

DBMS	RDBMS
1) DBMS applications store **data as file**.	RDBMS applications store **data in a tabular form**
2) In DBMS, data is generally stored in either a hierarchical form or a navigational form.	In RDBMS, the tables have an identifier called primary key and the data values are stored in the form of tables.
3) **Normalization is not** present in DBMS.	**Normalization is** present in RDBMS.
4) DBMS uses file system to store data, so there will be **no relation between the tables**.	in RDBMS, data values are stored in the form of tables, so a **relationship** between these data values will be stored in the form of a table as well.
5) DBMS has to provide some uniform methods to access the stored information.	RDBMS system supports a tabular structure of the data and a relationship between them to access the stored information.

6) DBMS **does not support distributed database**.	RDBMS **supports distributed database**.
7) DBMS is meant to be for small organization and **deal with small data**. it supports **single user**.	RDBMS is designed to **handle large amount of data**. it supports **multiple users**.
8) Examples of DBMS are file systems, **xml** etc.	Example of RDBMS are **mysql**, **postgre**, **sql server**, **oracle** etc.

Components of DBMS

A database system is composed of four components

1) Data
2) Hardware
3) Software
4) Users

which coordinate with each other to form an effective database system.

(1) **DATA :-** It is very important component of the database system. Most of the organizations generate, store and process large amount of data. The data acts as a bridge between the machine parts i.e. hardware and software and the users which directly access it or access it through some application programs. Data may be of different types.

(a) **User data** :- It consists of table(s) of data called relation(s) where Column(s) are called fields of attributes and rows are called Records for tables. A relation must be structured properly.

(b) **Metadata** :- A description of the structure of the database is known as metadata. It basically means "data about data". System Tables store the Metadata which includes :-

- Number of tables and tables name
- Number of fields and fields name
- Primary key fields

(2) Hardware :- The hardware consists of secondary storage devices such as magnetic disks (Hard Disk), optical disks (CD-ROM) ,on which data is stored together with the Input/Output devices (mouse, keyboard, printers), main memory

etc, which are used for storing and retrieving the data in a fast and efficient manner. Since database can range from those of a single user with a desktop computer to those on mainframe computers with thousands of users, therefore proper care should be taken for choosing appropriate hardware devices for a required database.

(3) Software :- Software part consists of DBMS which acts as a bridge between the user and the database or in other words, software that interacts with the users, application programs, and database and files system of a particular storage media (hard disk, magnetic tapes , etc) , to insert, update, delete and retrieve data. For performing these operations such as insertion, deletion and updation we can either use the Query

(4) Users :- Users are those persons who need the information from the database to carry out their primary business responsibilities i.e. Personnel , Staff, Clerical, Managers and Executives etc. On the basis of the job and requirements made by them , they are provided access to the database totally or partially.

The various types of users which can access the database are :-

(a) Database Administrators (DBA)
(b) Database Designers
(c) End Users
(d) Application Programmer

Database Administrator :- He/ She is a person who is responsible for the environmental aspects of a database. DBA is a person or group of persons who implements the policies of an organization. He is responsible for authorizing access monitoring database use, providing satisfactory response time, backup and recovery of the system failure. DBA has all the powers of database.

Database Designer :- He performs the duty to identify the data stored and also choose the database structure. he is also responsible to find out the relationship between the data, the constraints imposed on that data and for choosing database structure to store the data.

End Users :- End Users are those users who interact with the database through application program.

Application Programmer :- These are the professional users, who are responsible for writing the application program. The application program could be written in general purpose language such as COBOL , C.

What is Data Model

Data Model : A set of concepts to describe the structure of a database , and certain constraints that the database should obey .

Data Model Operations :-Operations for specifying database retrievals and updates by referring to the concepts of the data model . Operations on the data model may include basic operations and user defined operations.

Data Model Basic Building Blocks

(1) Entity :-

Anything about which data will be collected.

(2) **Attribute** :-

Characteristic of an entity.

(3) **Relationship** :-

Describes an association among entities.

(1) One to one (1:1) relationship

(2) One to many (1:M) relationship

(3) Many to many(M:M) relationship

(4) Constraint :-

A restriction placed on the data.

Importance of data models

Data Models :-

(a)Representations , usually graphical , of complex real world data structures.

(b) Facilitate interaction among the designer, the applications programmer and the end User .

End Users have different views and needs for data.

Hierarchical Data Model

It is the oldest type of data model, developed by IBM in 1968.This model organises data in a tree like structure, in which each child node can have only one parent node. the database based on the hierarchical data model comprises a set of records connected to one another through links. The link is an association between two or more records. The top of the tree structure consists of a single node that does not have any parent and is called the root node. Each parent can have many children but each child only has one parent. All attributes of a specific record are listed under an entity type. In a database, an entity type is the equivalent of a table, each individual record is represented as a row and an attribute as a column.

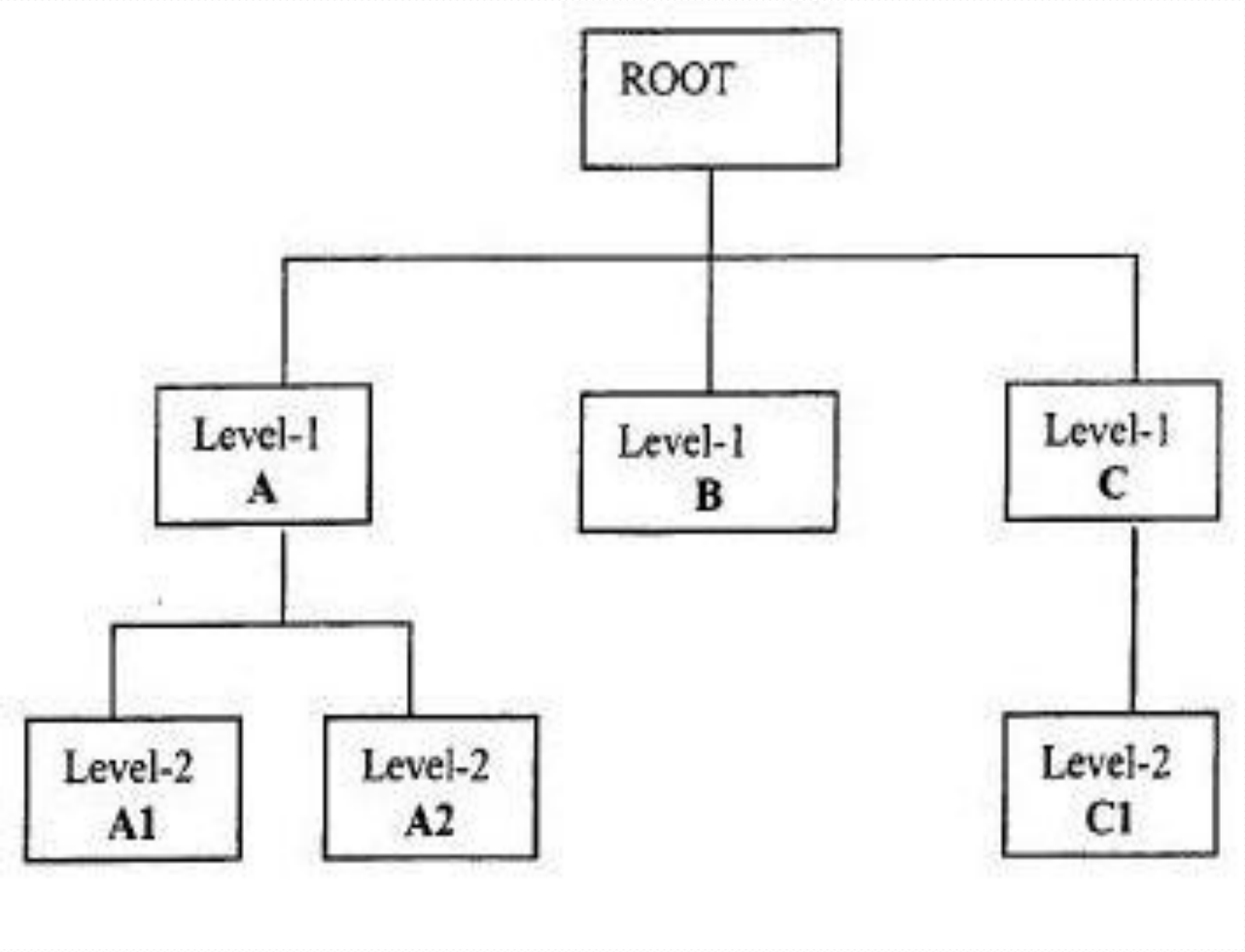
Salient Features :-

(1) Logically represented by an upside down tree.

(2) Each parent can have many children.

(3) The top layer is perceived as a parent of the segment directly beneath it.

(4) The segments below other segments are the children of the segment above them.


ROOT
Level-1
A
Level-1
B
Level-1
C
Level-2
A1
Level-2
A2
Level-2
C1

Advantages of Hierarchical Model

1) **Simplicity** :-Since the records are related in the form of parent-child relationship so performing the operations like insertion, deletion, etc on the tree like structure is very simple.
2) **Data integrity** :- As we know that the records are related through parent-child relationship. So each child can be accessed through its parent node or it can be reached through its parent node, so integrity of the data in this model is a major advantage.
3) **Data security** :- As we already know that each child node is linked to only one parent and it can be reached through its parent node so deleting any node needs to first search the parent node from the structure. this provides data security.

4) **Efficiency**:- Efficiency of this model is very high because this model handles the 1:N relationships between the parent-child records. When the database have bulk of records then this model handle the records efficiently.