**Chapter 1: Introduction to Database concepts**

**Database :** It is a collection of data. It contains information about one particular enterprise. E.g Bank, University, Company etc.

**Database Management System (DBMS)** : is a Collection of interrelated data and set of programs to access the data.

The objective of DBMS is to provide convenient and effective method of defining, storing and retrieving the information contained in the database.

In addition, the DBMS must provide for the safety of the information stored. It should protect the data from system crash or attempt at the unauthorized access. If the data are to be shared among several users, the system must avoid anomalous results. E.g. of DBMS are Oracle, Sqlserver, MySql, Access, Foxpro etc.

**File Processing System and Disadv. Of File Processing System :** Before DBMS the data is stored in the files. Different application programs are written to extract the data from these files and to add records to these files. But these system carries some disadv.

1. **Data Redundancy and Inconsistency :** The data files & application programs are created by different programmers over a long period of time
2. the data files has different formats.
3. Applications programs are written in different programming languages
4. The same information is may be duplicated in several files. This will results in data redundancy and inconsistency.

e.g. Consider following data files

*saving(account\_no,name,social\_security,address,telephone)*

*Current(account\_no,name,social\_security,address,telephone) Most of the fields are duplicated. i.e. Redundancy this results in increase in size and processing is slow*

*If the values of these common fields are not matching for some records in both files, then it results in the inconsistency of data.*

1. **Difficulty in accessing the data:** Conventional file processing system does not allow needed data to be retrieved in a convenient & efficient manner. E.g *saving(accno,name ,ssn,address,balance) Application programs to access the data are written. But if user wants to display those records for which balance >10000. And if the prog. Is not written, then it is difficult to access that data.*
2. **Data Isolation:** Because data is stored in different files & these files have different formats. It is difficult to write new application programs to retrieve the appropriate data.
3. **Integrity Problem :** The data values stored in the database must contains some consistency constraints e.g. balance must be >1000. Application programmer write the these constrains but when the time comes to change these constrains then its difficult to add the new constrains.
4. **Atomicity Problem :** Failures may leave database in an inconsistent state with partial updates carried out. E.g the following transaction Transfer of Rs. 500 from A’s Account to B’s Account

|  |  |
| --- | --- |
| *T1* | *Read (A)* |
|  | *A=A-500* |
|  | *Write (A)* |
| *T2* | *Read(B)* |
|  | *B=B+500* |
|  | *Write (B)* |

While the transaction is going on the Power Failure is occurs after the Transaction T1 the database will be in the inconsistent state. The All the transaction must happened all otherwise no transaction will be done.

1. **Concurrent Access Anomalies:** In case of file processing system, data is not centralized. If two or more users want to access the database at the same time interaction of concurrent updates may result in inconsistent data. E.g  **A is carring a Balance of Rs. 1000 and B is crediting Rs. 500 to A and C is withdraw Rs 500 from A’s account**

|  |  |
| --- | --- |
| *T1* | *T2* |
| *Read (A)* | *Read(A)* |
| *A=A+500* | *A=A-500* |
| *Write (A)* | *Write (A)* |

1. **Security Problem :** It is difficulty to provide authorized user access in the File processing system.

**Advantages of DBMS :**

1. **Centralized Management and Control over data :** DBA is overall control of Data and Users.
2. **Reduction of redundancies:** The DBMS contains some relation in the database it avoids the duplication of the data.
3. **Shared Data :** DBMS allows the sharing of data under its control by any number of application programmers and users.
4. **Integrity:** Centralized control also ensures that adequate checks are incorporated in the database to provide data integrity.
5. **Security :** DBA provides security by giving rights to the users.
6. **Data Independence :** DBA can change the structure of the data that will not affect the application program

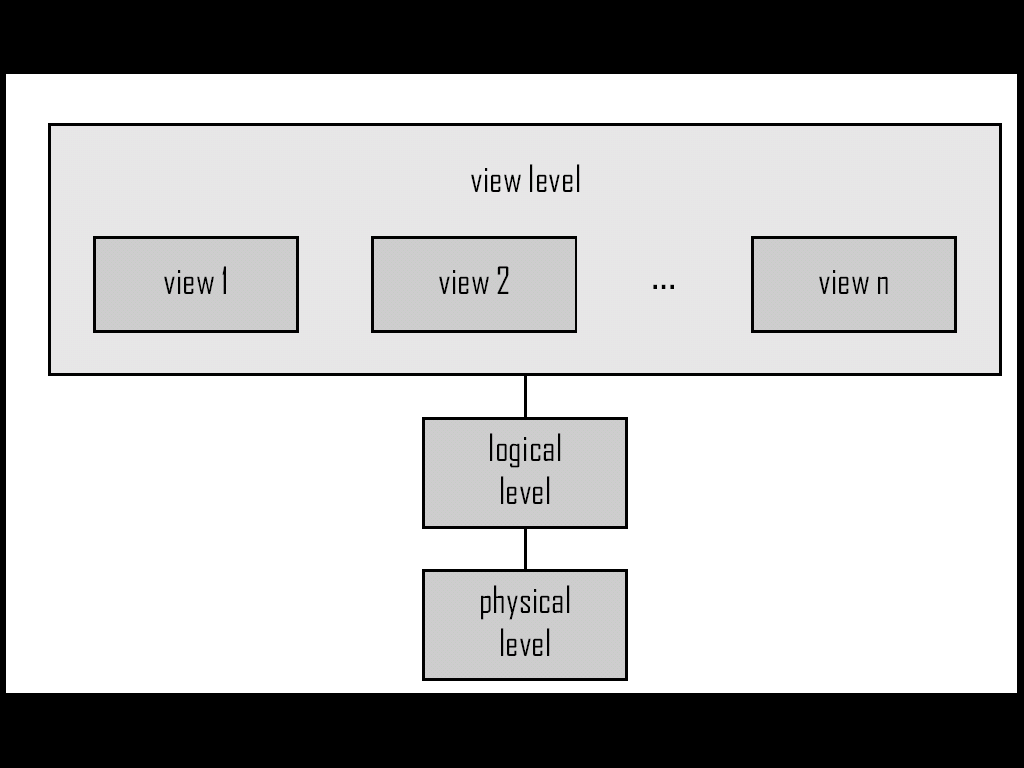
**Disadv. Of DBMS :**

1. As no. of users are associated with the centralization of data.
2. Cost of Software and Hardware.
3. Complexity of the back-up and recovery is increased.

**Data Abstraction :**  The DBMS is provides users with an abstract view of the data. Many db system users are computer trained persons, hence the complexity is hidden from them through several levels of abstraction.

**Defn** It means hides certain details of how the data is stored and maintained.

**Levels of Abstraction:**



**Physical Level**: It is lowest level of abstraction. It describes ***how*** the data is actually stored and describes the data structure and access methods to be used by database. (Database Creators)

**Logical Level (Conceptual Level)** :It describes ***what*** data is actually stored in the db and relationship exists among the data.(Insert) (DBA)

**View Level:** Itis the highest level of abstraction. It describes only a part of the entire db. Many users are not concern with the entire db they are interested in the part of db. To simplify their interaction with the db View level is defined. Application programs hide details of data types. Views can also hide information (such as an employee’s salary) for security purposes. (Application Programmner)

**e**.g

**type** *customer* = **record**

*customer\_id*:string;   
 *customer\_name*:string;  
 *customer*\_*street*:string;  
 *customer\_city* : string;

**end**;

The compliers reserves some memory for the variables it is physical level.

Programmers uses block this is called as Logical Level.

The users runs the programs and get the o/p this is view level.

**Schema of Database:** the logical structure of the database is called as schema of db (e.g., set of customers and accounts and the relationship between them).

According to the levels of abstraction the schema can be divided into two:

**Physical level schema:-** The overall structure of database at physical level.

**Logical level schema:-** The overall structure of database at logical level.

**Sub Schema**: The overall structure of db at view level. It is at the highest level.

**Data Independence:** The ability to modify a schema definition in one level without affecting a schema definition in the next higher level is called as data independence.

1. **Physical Data Independence**: It is the ability to modify the physical schema without causing application programs to be rewritten.
2. **Logical Data Independence :** It is the ability to modify the logical schema without causing application programs to be rewritten.

**Database Languages :**

1. **DDL:** To specify the database schema. The schema is specified by a set of definitions which are expressed by a special language called as DDL.
2. **Data Dictionary :** The result of compilation of DDL statement is a set of tables, which is stored in a special file called Data Dictionary or System Catalogue. This file contains **meta data** ie. data about data.
3. **Data Storage and Definition Language:** The storage structure and access methods used by the database system are specified by a set of definitions in a special type of DDL called data storage and Definition language.
4. **DML :** It enables the users to access or manipulate data as organized by the appropriate data model. i.e. Insert, Select ,Delete, Update. There are two types of the DML
5. **Procedural DML** : It requires a user to specify what data are needed and how to get those data.
6. **Non-procedural DML**: It requires a user to specify what data are needed without specifying how to get those data.

**Database Administrator**: The person having a central control over the system is called as DBA. He is responsible for the following things:

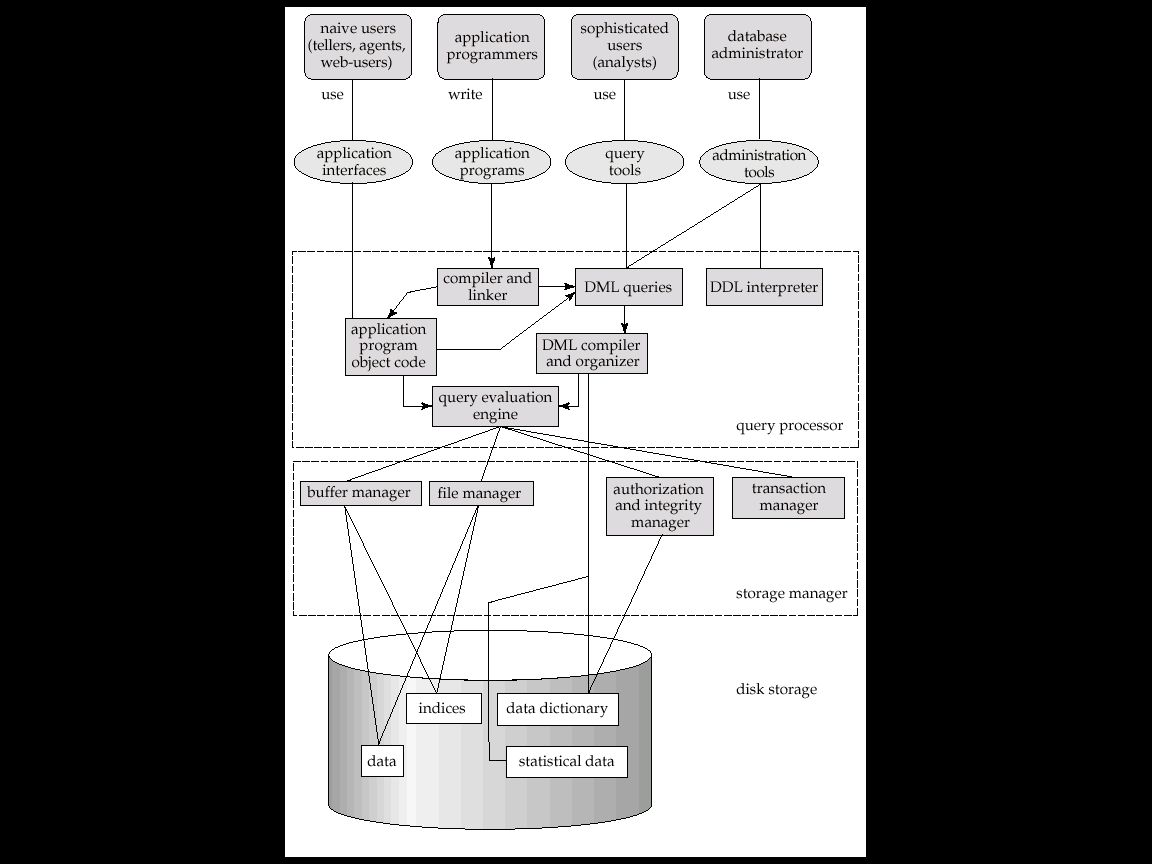
1. **Schema Definition**: These are written by DBA. These are translated by the DDL compiler to a set of tables that are permanently stored in the data dictionary.
2. **Storage Structure and Access Method definition:** The DBA creates appropriate storage structure and access methods by writing a set of definitions which is translated by data storage and data definition language compiler.
3. **Schema and Physical organization modification:** DBA writes a set of definitions to modify the database schema or description of physical storage organisation.
4. **Granting of authorization for data access:** DBA is responsible for granting the access to the Database.
5. **Integrity constraint specification:** Integrity constraints are written by DBA and they are stored in a special file, which is accessed by database manager while updating the data.

**Database Users:**

1. **Application Programmer:** These are computer professionals who are interact with the system through DML calls, which are embedded in a program written in a host language. These programs are commonly referred as application programs.
2. **Sophisticated Users:** They interact with the system through their requests written using a database query language. These requests are referred as Query.
3. **Specialized user:** They are sophisticated users who write specialized database applications that do not fit into the traditional data processing framework. E.g. Expert system, Knowledge system etc.
4. **Naïve Users:** They are unsophisticated users who interact with the system by involving one of the permanent application programs that have been written previously.

**Components of DBMS /Overall System Structure of DBMS :**  The system divided into

1. **Query Processor** :
2. **DML Compiler**: It translates DML statements in a query lang. into low level instructions that query evaluation engine understands. It also attempts to transforms users request into an equivalent but more efficient form.
3. **Embedded DML Pre-compiler**: It converts DML statements embedded in an application program to normal procedure calls in the host language.
4. **DDL Interpreter:** It interprets the DDL statements and records them in a set of tables containing meta data or data dictionary.
5. **Query Evaluation Engine**: It executes low-level instructions generated by the DML compiler.
6. **Storage Manager**: It will provide interface between the low-level data stored in the database and application programs and queries submitted to the system.
7. Authorization and Integrity manager: It will tests for the Authorization of data access & checks for integrity constraints.
8. Transaction manager: It ensures that the database remains in the consistent state despite the system failures and that concurrent transaction execution proceeds without conflicting.
9. File manager: It manages the allocation of the space on the disk storage and data structure used to represent information stored on disk.
10. Buffer Manager: It is responsible for fetching data from disk storage into main memory and deciding what data to cache in memory.
11. **Data Structure** :
12. Data Files: It stores the database.
13. Data Dictionary: It stores meta data about the structure of the database.
14. Indices: Provides fast access to data items that hold particular values.
15. Statistical Data : It stores statistical information about the data in the database. This information is used by query processor to select efficient ways to execute query.



**Concern when using an Enterprise Database:**  A centralized database provides online access to the database for many users concurrently. As the more no. of users are accessing the database risk is more. The issues are elaborated as follows:

1. **Enterprise vulnerability(Weakness):** If we store the data of an enterprise at the centralized one database often means that the database becomes more vital resource. The survival of the enterprise will depends on the information available 24X7. If the database is down no. body can work effectively in the enterprise.
2. **Confidentiality, privacy and security:** When the information is centralized and made available from the remote locations, it is possible to apply strict controls on who is able to see what information and who is able to modify what information. However the possibility of abuse can also increase because some hackers may be able to penetrate the security controls.
3. **Data quality** : Since a database provides its users the convenience to access information remotely, adequate controls are needed to control users updating data and thus control data quality and maintain integrity. With an increased no. of users accessing data directly, there is an increased risk that data quality and /or integrity is compromised partly because of input errors.
4. **The cost of building and using a DBMS:** The database approach provides a flexible approach to data management where new applications can be developed relatively inexpensively. The flexibility is not with the cost , but also with the training and operations of the enterprise.
5. **Inability to change the database when necessary:** If an enterprise wants to add the new business, change the existing business then database must be easily absorb the changes.

**Questions :**

1. **Describe different database users**
2. **Give and explain different advantages of DBMS over file processing system**
3. **Describe overall system architecture of DBMS with suitable diagram.**
4. **List four significant differences between file processing system and DBMS**
5. **Explain data dictionary storage.**
6. **Explain database schema.**
7. **Explain database abstraction and its types**
8. **Explain data independence and its types.**